

**MAGAZINE**

# **BSD**

**FOR NOVICE AND ADVANCED USERS**

## **Samba, NFS and Firewall LOCAL FILE SHARING**

**HARDENEDBSD,  
ALWAYS AHEAD  
IN SECURITY**

**HOW TO OVERRIDE  
PYTHON'S BUILT-IN  
METHODS AND TYPES**

**HOW TO USE  
DTRACE IN FREEBSD,  
OSX, SOLARIS,  
AND OPENSOLARIS**

**FREENAS CERTIFICATION**

VOL8 NO.12  
ISSUE 12/2014(65)  
1898-9144



855-GREP-4-IX  
[www.ixsystems.com](http://www.ixsystems.com)  
Enterprise Servers and Storage  
for Open Source



- ✓ Rock-Solid Performance
- ✓ Professional In-House Support

# FREENAS MINI STORAGE APPLIANCE

IT SAVES YOUR LIFE.



## HOW IMPORTANT IS YOUR DATA?

Years of family photos. Your entire music and movie collection. Office documents you've put hours of work into. Backups for every computer you own. We ask again, *how important is your data?*

## NOW IMAGINE LOSING IT ALL

Losing one bit - that's all it takes. One single bit, and your file is gone.

The worst part? **You won't know until you absolutely need that file again.**



*Example of one-bit corruption*

## THE SOLUTION

The FreeNAS Mini has emerged as the clear choice to save your digital life. **No other NAS in its class offers ECC (error correcting code) memory and ZFS bitrot protection to ensure data always reaches disk without corruption and never degrades over time.**

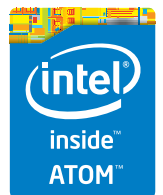
No other NAS combines the inherent data integrity and security of the ZFS filesystem with fast on-disk encryption. No other NAS provides comparable power and flexibility. The FreeNAS Mini is, hands-down, the best home and small office storage appliance you can buy on the market. **When it comes to saving your important data, there simply is no other solution.**

### The Mini boasts these state-of-the-art features:

- 8-core 2.4GHz Intel® Atom™ processor
- Up to 16TB of storage capacity
- 16GB of ECC memory (with the option to upgrade to 32GB)
- 2 x 1 Gigabit network controllers
- Remote management port (IPMI)
- Tool-less design; hot swappable drive trays
- FreeNAS installed and configured



<http://www.ixsystems.com/mini>



# FREENAS CERTIFIED STORAGE



With over six million downloads, FreeNAS is undisputedly *the* most popular storage operating system in the world.

Sure, you could build your own FreeNAS system: research every hardware option, order all the parts, wait for everything to ship and arrive, vent at customer service because it *hasn't*, and finally build it yourself while hoping everything fits - only to install the software and discover that the system you spent *days* agonizing over **isn't even compatible**. Or...

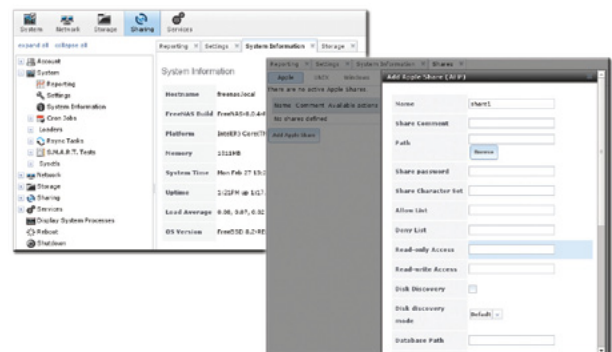
## MAKE IT EASY ON YOURSELF

As the sponsors and lead developers of the FreeNAS project, iXsystems has combined over 20 years of hardware experience with our FreeNAS expertise to bring you FreeNAS Certified Storage. **We make it easy to enjoy all the benefits of FreeNAS without the headache of building, setting up, configuring, and supporting it yourself.** As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS.

## Every FreeNAS server we ship is...

- » Custom built and optimized for your use case
- » Installed, configured, tested, and guaranteed to work out of the box
- » Supported by the Silicon Valley team that designed and built it
- » Backed by a 3 years parts and labor limited warranty

As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS. **Contact us today for a FREE Risk Elimination Consultation with one of our FreeNAS experts.** Remember, every purchase directly supports the FreeNAS project so we can continue adding features and improvements to the software for years to come. **And really - why would you buy a FreeNAS server from *anyone* else?**



### FreeNAS 1U

- Intel® Xeon® Processor E3-1200v2 Family
- Up to 16TB of storage capacity
- 16GB ECC memory (upgradable to 32GB)
- 2 x 10/100/1000 Gigabit Ethernet controllers
- Redundant power supply

### FreeNAS 2U

- 2x Intel® Xeon® Processors E5-2600v2 Family
- Up to 48TB of storage capacity
- 32GB ECC memory (upgradable to 128GB)
- 4 x 1GbE Network interface (Onboard) - (Upgradable to 2 x 10 Gigabit Interface)
- Redundant Power Supply



<http://www.iXsystems.com/storage/freenas-certified-storage/>

Dear Readers,

I hope you had a great time with your family during the Christmas holiday and now you are waiting for 2015. This time, I don't write what is inside this BSD issue and instead take advantage of the opportunity to wish you the next 365 days full of enjoyment, happiness, cheers, and unforgettable moments. May the New Year bring you more success, love and prosperity.

# HAPPY NEW YEAR 2015

And, I would like to thank You: our Readers; all iXsystems Company Employees who support BSD magazine and me, as without them publishing of the BSD magazine would not be possible; authors; reviewers; proofreaders; BSD fans; and friends for your invaluable support and contribution.

Enjoy reading,  
Ewa & the BSD Mag Team

## MAGAZINE BSD

**Editor in Chief:**

Ewa Dudzic  
ewa.dudzic@software.com.pl

**Contributing:**

Michael Shirk, Andrey Vedikhin, Petr Topiarz,  
Charles Rapenne, Anton Borisov, Jeroen van Nieuwenhuizen,  
José B. Alós, Luke Marsden, Salih Khan,  
Arkadiusz Majewski, BEng, Toki Winter, Wesley Mouedine  
Assaby, Rob Somerville

**Top Betatesters & Proofreaders:**

Annie Zhang, Denise Ebery, Eric Geissinger, Luca  
Ferrari, Imad Soltani, Olaoluwa Omokanwaye, Radjis  
Mahangoe, Mani Kanth, Ben Milman, Mark VonFange

**Special Thanks:**

Annie Zhang  
Denise Ebery

**Art Director:**

Ireneusz Pogroszewski

**DTP:**

Ireneusz Pogroszewski  
ireneusz.pogroszewski@software.com.pl

**Senior Consultant/Publisher:**

Paweł Marciniak  
pawel@software.com.pl

**CEO:**

Ewa Dudzic  
ewa.dudzic@software.com.pl

**Publisher:**

Hakin9 Media SK  
02-676 Warsaw, Poland  
Postepu 17D  
Poland  
worldwide publishing  
editors@bsdmag.org  
www.bsdmag.org

Hakin9 Media SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org.

All trademarks presented in the magazine were used only for informative purposes. All rights to trademarks presented in the magazine are reserved by the companies which own them.

# FreeNAS

## in an Enterprise Environment

By the time you're reading this, FreeNAS has been downloaded more than 5.5 million times. For home users, it's become an indispensable part of their daily lives, akin to the DVR. Meanwhile, all over the world, thousands of businesses, universities, and government departments use FreeNAS to build effective storage solutions in myriad applications.



### What you will learn...

- How TrueNAS builds off the strong points of the FreeBSD and FreeNAS operating systems
- How TrueNAS meets modern storage challenges for enterprise

The FreeNAS operating system is free to the public and offers thorough documentation, an active community, and a feature-rich storage environment. Based on FreeBSD, it can share over a host of protocols (SMB, FTP, iSCSI, etc) and features an intuitive interface, the ZFS file system, a plug-in system, and much more.

Despite the massive popularity of FreeNAS, many aren't aware of its big brother duties in some of the most demanding environments: the proven, enterprise-grade, professionally-supported line of TrueNAS.

But what makes TrueNAS different? Well, I'm glad you asked...

### Commercial Grade Support

When a mission critical storage solution goes down, an organization's whole operation can halt. Whole community-based (and free), it can't always get an immediate response and running in a timely manner. Responsiveness and expert support are critical. A dedicated support team can provide that safety.

Created by the same team that developed FreeNAS.

## WE INTERRUPT THIS MAGAZINE TO BRING YOU THIS IMPORTANT ANNOUNCEMENT:

THE PEOPLE WHO DEVELOP FREENAS, THE WORLD'S MOST POPULAR STORAGE OS, HAVE JUST REVAMPED TRUENAS.



## POWER WITHOUT CONTROL MEANS NOTHING. TRUENAS STORAGE GIVES YOU BOTH.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Simple Management   | <input checked="" type="checkbox"/> Self-Healing Filesystem            |
| <input checked="" type="checkbox"/> Hybrid Flash Acceleration                                 | <input checked="" type="checkbox"/> High Availability                  |
| <input checked="" type="checkbox"/> Intelligent Compression                                   | <input checked="" type="checkbox"/> Qualified for VMware and HyperV    |
| <input checked="" type="checkbox"/> All Features Provided Up Front (no hidden licensing fees) | <input checked="" type="checkbox"/> Works Great With Citrix XenServer® |

To learn more, visit: [www.iXsystems.com/truenas](http://www.iXsystems.com/truenas)



### POWERED BY INTEL® XEON® PROCESSORS

Intel, the Intel logo, Intel Xeon and Intel Xeon Inside are trademarks of Intel Corporation in the U.S. and/or other countries. VMware and VMware Ready are registered trademarks or trademarks of VMware, Inc. in the United States and other jurisdictions.

Citrix makes and you receive no representations or warranties of any kind with respect to the third party products, its functionality, the test(s) or the results therefrom, whether expressed, implied, statutory or otherwise, including without limitation those of fitness for a particular purpose, merchantability, non-infringement or title. To the extent permitted by applicable law. In no event shall Citrix be liable for any damages of any kind whatsoever arising out of your use of the third party product, whether direct, indirect, special, consequential, incidental, multiple, punitive or other damages.

# CONTENTS

## **DTrace, or Dynamic Tracing** **10**

**Carlos Neira**

DTrace, or dynamic tracing, was first available in Solaris 10 3/05 around 2005. DTrace is now available in FreeBSD beginning from 7.1 and Mac OS X from 10.5 (Leopard). DTrace differs from traditional tools in that code is instrumented dynamically (that means you can peek at the program without recompiling). You will learn how to use DTrace in FreeBSD, OSX, Solaris, and OpenSolaris.

## **Local File Sharing with Samba, NFS and Firewall** **16**

**Ivan Voras**

As opposed to how network file sharing is done in Windows, Samba works as an ordinary server process, similar to how the web server and other servers are implemented. There is nothing special about it and it is not significantly integrated into the operating system. You will learn more about the configuration of Samba, Windows file sharing protocols, as well as File sharing with NFS.

## **Python Flow Control Statements** **24**

**Pedro Araujo**

Let's start with basic control flow statements. The first statement will be the if/elif/else. This is the most basic control flow statement you can have in Python. You will learn how Python initialises new objects, how to override Python's built-in methods and types and how to get an instance's attributes using the Python shell.

## **HardenedBSD, Always Ahead in Security** **32**

**David Carlier**

Currently, FreeBSD uses the RC4 stream cipher for the arc4random family functions, both on the kernel and userland side. These functions serve many purposes; for example, on the kernel side, they allow the creation of proper randomized processes id, the stack protection canaries, and the HardenedBSD Address Space Randomization Layout uses them as well. You will learn more about the features of HardenedBSD.

## **Getting to Grips with the Gimp – Part 10** **34**

**Rob Somerville**

In the final part in the series on the Gimp, Rob will wrap up and take a look at how to further improve your Gimp experience.

## **“If you're moving information into the cloud, it just seems to me that all kinds of nasty activity could go on in there. I would take a Missouri approach and say – prove it to me, show it to me – how it's more secure”.** **38**

**Rob Somerville**

# Among clouds Performance and Reliability is **critical**



Download syslog-ng Premium Edition  
product evaluation [here](#)

Attend to a free logging tech webinar [here](#)



**BalaBit**  
IT Security

[www.balabit.com](http://www.balabit.com)

## **syslog-ng log server**

The world's first High-Speed Reliable Logging™ technology

### **HIGH-SPEED RELIABLE LOGGING**

- above 500 000 messages per second
- zero message loss due to the  
Reliable Log Transfer Protocol™
- trusted log transfer and storage

# FreeNAS Certification

## Why?

The software defined storage (SDS) market is growing rapidly. Customers are implementing a software-defined data center (SDDC) and find that storage is the last large component of the datacenter that they need to implement. Many of these customers are looking at FreeNAS, which is an open source SDS that provides enterprise NAS/SAN storage using commodity hardware. FreeNAS is the industry leader in SDS and is seen as a vital part of SDDC. With over 7 million downloads, 100s of contributors, 20+ releases, almost 300K lines of code, and the support of multiple corporations, the demand for this product is massive and growing.

The storage market is growing around 50% annually. Recent studies have shown that storage is the largest cost of workload deployment. With its massive scaling capabilities, compelling economics, and disk drive densities rapidly increasing, FreeNAS is an economical solution

The 5 classes build on each other. All 5 are required for certification.

## Intro to FreeNAS

In the first class we start with the core skills. The student will learn to build a basic pool and share datasets using CIFS/SMB, AFP, and NFS for physical and virtual applications.

The next class strengthens these core skills and adds System Administration and Storage Administration. The student will learn administrative tools for ongoing management of a FreeNAS System, like upgrade, disk replacement and repair. We also cover disaster recovery in detail.

Next we dive deep into sharing. We make sure students understand how the different protocols work and how to set-up, custom configure, and troubleshoot each of the protocols SMB/CIFS, AFP, and NFS. We add an in-depth look at iSCSI.

The next class covers Hardware architecture. There are a large number of enterprise use cases for FreeNAS and

# FreeNAS<sup>®</sup>

that can scale systems to multiple petabytes without having to partition workloads.

## Goal

To build a crowd sourced team of enterprise certified professionals to handle the growing demand for consulting and support services for FreeNAS.

## Audience

The FreeNAS community includes the hobbyist/home user and companies that use FreeNAS as a commercial/enterprise deployed SDS. The forums and community serves the hobbyist quite well, but the enterprise customer also requires training, support, and consulting.

## Certification Process

5 Classes. The intro class is free!

You can learn more by going to <http://www.freenas.org/freenas-zfs-training/>.

we discuss different components and configurations for several of these. We also look at common tunables for these different workloads.

The final class finishes with Advanced Administration. This class covers common tools for debugging problems and configuration of tools including Active Directory, Network Services, and Jails. This class also covers command line ZFS tools and advanced troubleshooting.

## Exam

After all five of the classes are attended, then you can take a certification exam. The certification is provided in person and online. Candidates must demonstrate basic skills as covered in class one, as well as answer questions from the remaining classes. Questions on the basic skills require a 100% score and the answers on the remaining questions require an 80% score to receive certification. The test can be retaken as many times as needed.

UPDATE  
NOW WITH  
**STIG**  
AUDITING

“IN SOME CASES  
**nipper studio**  
HAS VIRTUALLY  
**REMOVED**  
the **NEED FOR** a  
**MANUAL AUDIT**”  
CISCO SYSTEMS INC.

Titania's award winning Nipper Studio configuration auditing tool is helping security consultants and end-user organizations worldwide improve their network security. Its reports are more detailed than those typically produced by scanners, enabling you to maintain a higher level of vulnerability analysis in the intervals between penetration tests.

Now used in over 45 countries, Nipper Studio provides a thorough, fast & cost effective way to securely audit over 100 different types of network device. The NSA, FBI, DoD & U.S. Treasury already use it, so why not try it for free at [www.titania.com](http://www.titania.com)



[www.titania.com](http://www.titania.com)

# DTrace, or Dynamic Tracing

DTrace, or dynamic tracing, was first available in Solaris 10 3/05 around 2005. DTrace is now available in FreeBSD beginning from 7.1 and Mac OS X from 10.5 (Leopard). DTrace differs from traditional tools in that code is instrumented dynamically (that means you can peek at the program without recompiling).

---

## What you will learn...

- How to use DTrace in FreeBSD, OSX, Solaris, and OpenSolaris

## What you should know...

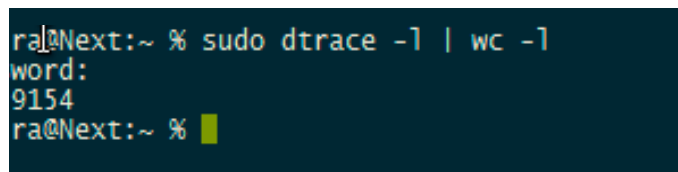
- FreeBSD basics
- 

From the handbook: [https://www.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/dtrace.html](https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/dtrace.html).

“The FreeBSD implementation provides full support for kernel DTrace and experimental support for userland DTrace. Userland DTrace allows users to perform function boundary tracing for userland programs using the pid provider, and to insert static probes into userland programs for later tracing. Some ports, such as databases/postgres-server and lang/php5 have a DTrace option to enable static probes. FreeBSD 10.0-RELEASE has reasonably good userland DTrace support, but it is not considered production ready. In particular, it is possible to crash traced programs.”

## Requirements

I'm running FreeBSD 10.0-STABLE where DTrace is already available as a kernel module. Typing the following as root will let you know that you are ready to fire up some probes using DTrace: Figure 1.

A terminal window with a dark background. The prompt is 'ra@Next:~ %'. The user enters 'sudo dtrace -l | wc -l'. The output is '9154'. The prompt returns to 'ra@Next:~ %' with a green cursor.

```
ra@Next:~ % sudo dtrace -l | wc -l
9154
ra@Next:~ %
```

Figure 1.

If this fails, you need to recompile your kernel and follow the instructions from the handbook in here: [https://www.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/dtrace-enable.html](https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/dtrace-enable.html).

## Why do I care about DTrace?

If you want to understand what is happening in your software without needing recompiled special versions of your applications (lots of debug messages, maybe recompile with debug flags to use a debugger?) and also centralize all your current instrumentation tools into just one, then you should care about DTrace.

Some features, not all:

- DTrace is dynamic: probes are enabled only when you need them
- No code is present for inactive probes
- There is no performance degradation when you are not using DTrace
- When the DTrace command exits, all probes are disabled and instrumentation removed
- The system is returned to its original state
- DTrace is nondestructive. The system is not paused or quiesced
- DTrace is designed to be efficient. No extra data are ever traced

- Because of its safety and efficiency, DTrace can be used in production to solve real problems in real time
- Predicates: A logical predicate mechanism allows actions to be taken only when user-specified conditions are met. Unwanted data is discarded at the source – never retained, copied, or stored
- A high-level control language: DTrace is equipped with an expressive C-like scripting language known as D. It supports all ANSI C operators, which may be familiar to you and reduce your learning curve, and allows access to the kernel's variables and native types. D offers user-defined variables, including global variables, thread-local variables, and associative arrays, and it supports pointer dereferencing.

## Trying DTrace

We will try a default script that comes with our FreeBSD installation. Go to `/usr/share/dtrace/toolkit` and execute the script called `procsystime`; the script “only process system call time details.” Notice the only (Figure 2).

This one is pretty handy right away; imagine what you could do with some imagination.

```
root@bsd:/usr/share/dtrace/toolkit # ./procsystime
Tracing... Hit Ctrl-C to end...
^C
Elapsed Times for all processes,

SYSCALL          TIME (ns)
sigreturn         5778
sigaction         12857
fstat             15745
getpid            18541
__sysctl          39645
munmap            47318
getsockopt        51057
mmap              58251
read              97469
sigprocmask       278332
ioctl             474094
clock_gettime     683811
write             1123755
_umtx_op          3011901084
select            3014173473

root@bsd:/usr/share/dtrace/toolkit #
```

Figure 2.

Let's see this one liner: which processes are executing the most system calls? (Figure 3)

Pretty impressive. At this moment you are wondering how all these years you have lived without DTrace. Using `truss`, `strace`, `lsof`, even `gdb` seems pretty lame now, well, `gdb` is not so lame now.

## DTrace Scripting

DTrace scripts are written in the D language; you could take a look at this reference <http://dlang.org/spec.html>.

Now let's write our first probe. A DTrace script has the following structure:

```
Your probes
/ predicate (usually you will create a filter here) /
{
    What are you going to do when you hit a probe
}
```

Let's create a simple one to get used to the syntax, and we will dissect it line by line. This one does not have a predicate so it will capture all that the probe is asking for.

A predicate is a conditional statement (IF statement, if you like; see Figure 4).

Save this to a file called `example1.d`, then execute the script typing:

```
dtrace -s example1.d
```

The probe section has the following syntax:

```
provider:module:function:name
```

```
syscall:::entry ➡ your probe
{
    @[pid,execname] = count(); ➡ your action
}
```

Figure 4.

```
root@bsd:/usr/share/dtrace/toolkit # dtrace -n 'syscall:::entry { @[pid, execname] = count();}'
dtrace: description 'syscall:::entry ' matched 536 probes
^C
1398  preload          4
1377  sendmail         24
2118  sshd             32
2273  dtrace           233

root@bsd:/usr/share/dtrace/toolkit #
```

Figure 3.

Table 1.

Provider	The name of the DTrace provider that is publishing this probe. The provider name typically corresponds to the name of the DTrace kernel module that performs the instrumentation to enable the probe.
Module	If this probe corresponds to a specific program location, the name of the module in which the probe is located. This name is either the name of a kernel module or the name of a user library.
Function	If this probe corresponds to a specific program location, the name of the program function in which the probe is located.
Name	The final component of the probe name is a name that gives you some idea of the probe's semantic meaning, such as BEGIN or ENDS in this case the probe says that is the entry of a function call.

What providers we have available in FreeBSD ? well you should dig in and see what you need.

```
root@bsd:~/dtracescripts # dtrace -l | head -1
ID PROVIDER MODULE FUNCTION NAME
root@bsd:~/dtracescripts # dtrace -l | head -2
ID PROVIDER MODULE FUNCTION NAME
1 dtrace BEGIN
root@bsd:~/dtracescripts # dtrace -l | head -5
ID PROVIDER MODULE FUNCTION NAME
1 dtrace BEGIN
2 dtrace END
3 dtrace ERROR
4 fbt kernel camstatusentrycomp entry
root@bsd:~/dtracescripts # dtrace -l | head -7
ID PROVIDER MODULE FUNCTION NAME
1 dtrace BEGIN
2 dtrace END
3 dtrace ERROR
4 fbt kernel camstatusentrycomp entry
5 fbt kernel camstatusentrycomp return
6 fbt kernel cam_compat_handle_0x17 entry
root@bsd:~/dtracescripts #
```

Figure 5.

```
root@bsd:~/dtracescripts # dtrace -s example1.d
dtrace: script 'example1.d' matched 536 probes
^C

1466 sh 5
1244 syslogd 10
1464 cron 20
1467 preload 22
1376 sendmail 24
1397 preload 54
1466 dd 67
1467 sh 88
1465 cron 149
1467 vmstat 237
1465 sh 282
1383 cron 2374
root@bsd:~/dtracescripts # cat example1.d
syscall::entry
/execname != "dtrace" /
{
    @[pid,execname] = count();
}
```

Figure 6.

Table 2.

Function Name	Arguments	Result
count	none	The number of times called.
sum	scalar expression	Linear total value of the specified expressions.
avg	scalar expression	The arithmetic average of the specified expressions.
min	scalar expression	The smallest value among the specified expressions.
max	scalar expression	The largest value among the specified expressions.
lquantize	scalar expression, lower bound, upper bound, step value	A linear frequency distribution, sized by the specified range, of the values of the specified expressions. Increments the value in the highest bucket that is less than the specified expression.
quantize	scalar expression	A power-of-two frequency distribution of the values of the specified expressions. Increments the value in the highest power-of-two bucket that is less than the specified expression.

What every section means Table 1.

Now this line:

```
@[pid,execname] = count();
```

This is called an aggregation and is denoted by the @ special character. Aggregations are global in your DTrace scripts. The syntax for an aggregation is the following:

```
@name[ keys ] = aggfunc ( args );
```

a d v e r t i s e m e n t



better safe than sorry  
www.demyo.com

Name: The name you choose for the aggregation.

Keys: Comma-separated list of D expressions (in this case, we are asking for pid and name of executable triggering the probe). Aggfunc: Is one of the DTrace aggregating functions, and args is a comma-separated list of arguments appropriate for the aggregating function.

Here are the aggregation functions available: Table 2.

Now, let's add a predicate to the same script. If you looked at the output of the script, it also counted the systemcalls done by DTrace itself. Let's filter that (Figure 6).

But how did I know that execname contained the name of the program being executed? Well, it is a built in variable in DTrace. Here is a list of some of them, and you can take a look at the full listing at this url [http://docs.oracle.com/cd/E18752\\_01/html/819-5488/gcfpz.html](http://docs.oracle.com/cd/E18752_01/html/819-5488/gcfpz.html).

## Conclusion

We approached both situations using DTrace, which is available in FreeBSD, OSX, Solaris and opensolaris and checked if this tool is beneficial and a time saver in the process.

---

## ABOUT THE AUTHOR

*Carlos Neira has worked several years as a C/C++ developer and kernel porting and debugging enterprise legacy applications. He is currently employed as a C developer under Z/OS, debugging and troubleshooting legacy applications for a global financial company. Also, he is engaged in independent research on affective computing. In his free time, he contributes to the PC-BSD project and enjoys metal detecting.*

**Table 3.** DTrace Built. Variables

Type and Name	Description
int64_t arg0, ..., arg9	The first ten input arguments to a probe represented as raw 64-bit integers. If fewer than ten arguments are passed to the current probe, the remaining variables return zero.
args[]	The typed arguments to the current probe, if any. The <code>args[]</code> array is accessed using an integer index, but each element is defined to be the type corresponding to the given probe argument. For example, if <code>args[]</code> is referenced by a <code>read(2)</code> system call probe, <code>args[0]</code> is of type <code>int</code> , <code>args[1]</code> is of type <code>void *</code> , and <code>args[2]</code> is of type <code>size_t</code> .
uintptr_t caller	The program counter location of the current thread just before entering the current probe.
string cwd	The name of the current working directory of the process associated with the current thread.
uint_t epid	The enabled probe ID (EPID) for the current probe. This integer uniquely identifies a particular probe that is enabled with a specific predicate and set of actions.
int errno	The error value returned by the last system call executed by this thread.
string execname	The name that was passed to <code>exec(2)</code> to execute the current process.
gid_t gid	The real group ID of the current process.
uint_t id	The probe ID for the current probe. This ID is the system-wide unique identifier for the probe as published by DTrace and listed in the output of <code>dtrace -l</code> .
lgrp_id_t lgrp	The latency group ID for the latency group of which the current CPU is a member.
pid_t pid	The process ID of the current process.
pid_t ppid	The parent process ID of the current process.
string probefunc	The function name portion of the current probe's description.
string probemod	The module name portion of the current probe's description.
string probename	The name portion of the current probe's description.
string probeprov	The provider name portion of the current probe's description.
uint64_t timestamp	The current value of a nanosecond timestamp counter. This counter increments from an arbitrary point in the past and should only be used for relative computations.
uid_t uid	The real user ID of the current process.
uint64_t uregs[]	The current thread's saved user-mode register values at probe firing time. Use of the <code>uregs[]</code> .
uint64_t vmregs[]	The current thread's active virtual machine register values at probe firing time. Use of the <code>vmregs[]</code> .
uint64_t vtimestamp	The current value of a nanosecond timestamp counter that is virtualized to the amount of time that the current thread has been running on a CPU, minus the time spent in DTrace predicates and actions. This counter increments from an arbitrary point in the past and should only be used for relative time computations.
uint64_t walltimestamp	The current number of nanoseconds since 00:00 Universal Coordinated Time, January 1, 1970.



# Dr.Web 9.0

## for Windows — the rapid response anti-virus

1. Reliable protection against the threats of tomorrow
2. Reliable protection against data loss
3. Secure communication, data transfer and Internet search



© Doctor Web  
2003 — 2013

[www.drweb.com](http://www.drweb.com)

**Free 30-day trial:** <https://download.drweb.com>

**New features in Dr.Web 9.0 for Windows:** <http://products.drweb.com/9>

**FREE bonus — Dr.Web Mobile Security:**  
<https://download.drweb.com/android>



# Local File Sharing with Samba, NFS and Firewall

As opposed to how network file sharing is done in Windows, Samba works as an ordinary server process, similar to how the web server and other servers are implemented. There is nothing special about it and it is not significantly integrated into the operating system.

---

## What you will learn...

- The basic configuration of Samba.
- Windows file sharing protocols.
- File sharing with NFS.

## What you should know...

- Unix-like systems basics.

There are significant differences in how Windows and Unix-like systems handle files, and one of those is how users, groups and file permissions are integrated. In Unix-like systems, users and groups are directly represented by numbers called the “user ID” and the “group ID” (often abbreviated as UID and GID). Files are owned by a single user and a single group, and they have a uniform permission mask which specifies if the file can be read, written or executed by the user, the group, or anyone else. It’s a robust and simple model. In Windows, both users and groups (as well as other resources on the system) are represented by “security identifiers” (SID), which are long sequences of almost-random numbers. Files have a large number of possible permissions which can be allowed or denied to any specific SID.

Another difference is that Windows treats file and user names as case-insensitive, while Unix-like systems are, of course, fully case-sensitive.

All these differences (and there are several more) mean that Samba needs to provide a significant amount of emulation and adaptation between what the data is on the

Unix-like system and what is presented to the Windows clients. This emulation is not perfect, so in practical operation the systems which use Samba extensively follow neither traditional Unix semantics nor are completely compatible with Windows.

### Command Line Information

Command line may be prefixed with a # to indicate they need to be executed as a “root” user, or a > to indicate they can be executed as a regular non-privileged user. Those will be set in `monospaced font` for distinctiveness and readability.

## Installing Samba

The entire fourth major version of Samba is relatively recent and it has introduced significant changes in its internal operation, in order to support emulating Active Directory servers. However, setting up an Active Directory server with Samba is an advanced topic. We will describe how to use Samba for simple file sharing in a relatively small office workgroup.

The most recent version of Samba available as a package is Samba 4.1, but it doesn't contain the `pam_smbpass` module, so the recommended version to use is 4.0, available in the "samba4" package.

### Initial configuration

Samba is configured by a single configuration file, `/usr/local/etc/smb4.conf`, which can contain a huge amount of options, documented in the `smb4.conf(5)` main page.

As a starting point, it can contain these basic options:

```
[global]
workgroup = MYOFFICE
server string = My Office Server
dos charset = cp852
unix charset = UTF-8
security = user
encrypt passwords = yes
socket options = TCP_NODELAY SO_RCVBUF=262144
SO_SNDBUF=262144
use sendfile = yes
use mmap = yes
unix extensions = no
wide links = yes

[homes]
comment = Home Directories
browseable = no
writeable = yes
directory mask = 0770
create mask = 0660

[public]
comment = Public
path = /srv/public
public = no
writeable = yes
write list = @wheel
directory mask = 0770
create mask = 0660
```

The file is structured similar to Windows .INI files. There are a few special sections, and one of these is `[global]`.

This section contains global configuration options, and those in the above code are:

- `workgroup` – the name of the Windows ("Network neighbourhood") workgroup
- `server string` – the server description shown in Windows Explorer
- `dos charset` – specify which character sets will be used by non-unicode applications on the client side

- `unix charset` – specifies which character set will be used on the server to store file names
- `security` – the type of the server. There are several security types which can be used, but the "user" type most closely resembles a desktop version of Windows (e.g., Windows XP, Windows 7). Other options include "share" which resembles Windows 95, "domain" if the system will be a part of a Windows NT 4-style domain, or "ads" if it will be a part of an Active Directory domain
- `encrypt passwords` – forces clients to transmit only encrypted passwords over the network (to prevent old or malicious clients from connecting to the server)
- `socket options` – sets network socket optimizations; specifically, it disables Nagle's algorithm, and sets larger buffer sizes than would be by default
- `use sendmail`, `use mmap` – activates some generic optimizations in file access and serving
- `unix extensions`, `wide links` – unix extensions are only useful to unix-like clients, not Windows; wide links allows the Samba server to follow any symbolic links in the file system as if they were ordinary files and directories, which is the behaviour expected by the users; enabling both unix extensions and wide links can enable remote clients to create and access any file on the server they have the appropriate permissions to access by creating symlinks (instead only the "shared" files), which is why these options are somewhat controversial.

Another special section is `[homes]`, which configures how the traditional Unix-like home directories will be automatically shared with Samba. This section can contain any configuration options available for configuring normal shares (like the `[public]` section does), only the settings will apply to all home directories, which will be accessible as separate network shares named `\\SERVER\username`.

The specific options used here are:

- `comment` – the share description, visible in Windows Explorer
- `browseable` – will the share (or, in this special case, all home directories which will be presented as shares) be visible in Windows Explorer when the top-level server's list of shares is accessed (the address `\\SERVER`)
- `writeable` – will any users, under any circumstances (file permissions, etc) be able to write to this share
- `directory mask` – when new directories are created, they will be owned by the user which creates them and their Unix group will be set to the default group of this

user, but the Unix file permissions will be set to the traditional octal number mask given in this directive

- create mask – similar to directory mask, but for ordinary files.

The [public] section, finally, configures an ordinary share, available in Windows Explorer as `\\SERVER\public`. We will configure this share to be accessible to all users of the system group “wheel”, and we will place it in the `/srv/public` directory. The additional configuration options (not described previously) are:

- path – the directory which will be shared
- public – will the share be accessible without a password
- write list – a list of users or groups (group names are prefixed with “@”) which will be able to write to this share (the users still need to be given adequate permissions inside the directory itself).

After creating `smb4.conf`, Samba needs to be enabled by adding the following line in:

```
/etc/rc.conf:
samba_server_enable="YES"
```

## Synchronizing users

In order to function optimally, Samba needs to map users which Windows clients use to system users. This means that for each Windows user Samba is used with, there needs to be a Unix-like system user to match it. To complicate things even more, Unix-like systems and Windows systems store passwords in completely incompatible formats.

This can be resolved in three ways: firstly, by manually maintaining FreeBSD users and synchronizing it with a Samba-specific user database by using the `smbpasswd` utility. This utility behaves similarly to the “pw” utility, or as a combination of `adduser` / `deluser` / etc. utilities. This is the simplest solution and is recommended for small installations with several users which very rarely change their passwords. Note that you will first need to add a FreeBSD user, and then add a Samba user with the same name with

```
"smbpasswd-a"
```

The second way is to automatically replicate the two password databases. In this scenario, both the FreeBSD user database and the Samba user database exist at the same time, but the passwords are copied from one to the other as needed. Add the following line:

```
pam password change = yes
```

to the [global] section of `smb4.conf` to enable users to change the password from Windows, which Samba will propagate to the FreeBSD system users database. The other direction (changing the system user password and propagating it to the Samba users database) can be enabled by adding the following line to the `/etc/pam.d/system` file:

```
auth optional
/usr/local/lib/pam_smbpass.so migrate
```

and the following line to the `/etc/pam.d/passwd` file:

```
password required
/usr/local/lib/pam_smbpass.so nullok use_authtok try_first_pass
```

This will enable users to change both their FreeBSD password and Samba password at the same time using the “passwd” utility, as well as “migrate” (copy) their FreeBSD password to the Samba database when they next log in.

Note that you still need to maintain group memberships outside Samba, in the `/etc/group` file, and that the users still need to be created (with the `-a` argument to `smbpasswd`) in the Samba database.

The third way is to have the primary user database only in Samba, and use the Winbind facility of Samba to make those users available to the system via the PAM and NSS mechanisms. In this way, you do not need to create or maintain FreeBSD users and groups, only Samba users and groups, and FreeBSD will treat them as if they were local system users.

## File permissions

The `smb4.conf` file regulates file access on a very coarse basis – it only lists the users which may access the Windows share. After this security check passes, the users are allowed file system operations based on what permissions they have on the specific file system objects.

By default, only regular Unix permissions are checked, and any attempt to set additional Windows permissions on the client side (e.g. through the Windows Explorer) are ignored, or may even produce unwanted results. If you do not do any of the following steps, you should only regulate file access permissions directly from FreeBSD by using the usual `chown` / `chgrp` / `chmod` commands.

Windows-style ACL lists can be introduced by adding the mount option “`nfsv4acls`” to the desired file system, and enabling the “`zfsacl`” VFS plugin in `smb4.conf`.

The `fstab` entry should look like this:

```
/dev/da0p2 / ufs rw,nfs4acls 1 1
```

And the additional configuration entries for the global section in `nfs4.conf` are:

```
vfs objects = zfsacl
map acl inherit = Yes
inherit permissions = Yes
inherit acls = Yes
store dos attributes = Yes
```

However, the UFS file system does not fully support permission inheritance which is normal Windows ACL semantics. Samba can get around this limitation with the “map acl inherit” option, but such functionality will not be available to regular FreeBSD applications.

## File sharing with NFS

In contrast to Samba, NFS, on most Unix-like systems, is served by a kernel process instead of a userland process. This is a consequence of historical performance issues that have carried into modern systems that do not have as many problems with context switching and network I/O. To complicate things further, its configuration is driven from userland, and from a generic RPC service. NFS requires the interaction of several different parts, and if any one of them fails, strange errors may occur.

Firstly, there is the `rpcbind` service (previously known as the “portmapper”), which acts as a broker between remote clients and local RPC servers. It needs to be enabled as a prerequisite for everything else. Next, the `mountd` service (one of those RPC services) accepts remote file system mount requests, checks local file system export and security settings, and enables NFS sharing if everything passes. The NFS server itself is actually two servers – one for old NFS versions 2 and 3, and one for NFS version 4. Additionally, NFSv3 requires two more services to perform operations not standardised in the core protocol: the `statd` and the `lockd` services, which provide persistent file status and cross-client locking operations. Without the latter two services, NFS is completely stateless and file locks are not visible across different remote clients. The complications listed above are not FreeBSD-specific.

As opposed to Samba, NFS has a somewhat weak authentication mechanism. The basic operation of NFS is to share specific directory trees to specific IP addresses, without any additional authentication. This means that, if a situation arises where an IP address can be forged, the files could be accessed by undesirable parties.

In versions 2 and 3, NFS relies on regular Unix-like file system permissions literally, using only the integer user

**The BSD Certification Group Inc. (BSDCG) is a non-profit organization committed to creating and maintaining a global certification standard for system administration on BSD based operating systems.**

## ? WHAT CERTIFICATIONS ARE AVAILABLE?

**BSDA: Entry-level certification** suited for candidates with a general Unix background and at least six months of experience with BSD systems.

**BDSP: Advanced certification** for senior system administrators with at least three years of experience on BSD systems. Successful BDSP candidates are able to demonstrate strong to expert skills in BSD Unix system administration.

## ✓ WHERE CAN I GET CERTIFIED?

**We're pleased to announce that after 7 months of negotiations and the work required to make the exam available in a computer based format, that the BSDA exam is now available at several hundred testing centers around the world. Paper based BSDA exams cost \$75 USD. Computer based BSDA exams cost \$150 USD. The price of the BDSP exams are yet to be determined.**

Payments are made through our registration website:  
<https://register.bsdcertification.org/register/payment>

## i WHERE CAN I GET MORE INFORMATION?

More information and links to our mailing lists, LinkedIn groups, and Facebook group are available at our website:  
<http://www.bsdcertification.org>

Registration for upcoming exam events is available at our registration website:  
<https://register.bsdcertification.org/register/get-a-bsdcg-id>

IDs and group IDs as signals of permissions and ownerships. This means that, if two different users on two different machines have the same UID, they will be able to access each other's files over NFS. Consequently, if a person has two accounts on different systems (e.g. the server and the client) with different UIDs, he will not be able to access his own files. The problems only increase with the existence of user groups. There are some workarounds for this problem, including using the NIS system to synchronize user (and group) account data across different machines (supported on FreeBSD), and using an "idmap" service which performs user ID re-mapping (not supported on FreeBSD).

NFS version 4 somewhat reduces this problem by transferring user (and group) names instead of UIDs over the network, but user information (their accounts and group membership information) still needs to be synchronised externally. Modern setups may accomplish this with either NIS, Kerberos, LDAP, WinBind, or even Active Directory.

### Configuring the NFS services startup

To fully configure NFS with protocol versions 2, 3 and 4, and with all of the services listed in the previous section, the following lines are required in `/etc/rc.conf`:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
nfsv4_server_enable="YES"
nfsuserd_enable="YES"
nfsuserd_flags="-domain example.com"
nfs_reserved_port_only="YES"
rpc_statd_enable="YES"
rpc_lockd_enable="YES"
mountd_enable="YES"
mountd_flags="-p 801"
```

In addition to enabling the described services, the configuration lines also enable the "nfsuserd" service which helps NFSv4 get a list of usernames into the kernel, instructs it to consider the users a part of the "example.com" domain, and forces the NFS server and the mountd service to use well-known, specific ports.

### Exporting file system trees over NFS

The list of directories with permissions is maintained in the `/etc/exports` file. This is a text file which usually has one line per exported file system, except a single special line which configures NFSv4. An example file content may be:

```
V4: / -sec=sys
/home -alldirs -network 192.168.1.0 -mask 255.255.255.0
```

```
-maproot 0
/data/distfiles -alldirs -network 192.168.1.0 -mask
255.255.255.0 -maproot 0
/usr/ports -alldirs -network 192.168.1.0 -mask
255.255.255.0 -maproot 0
```

The first line in the above code configures NFSv4 so that it allows sharing our entire file system (i.e. rooted in the "/"), and that security will be a direct extension of traditional Unix-like security with users in the system user database ("sys").

The next three lines export three directories, allowing mounting any of their subdirectories to the network 192.168.1.0/24, and maps the remote root user (which always has UID=0) to the local root user.

This last part exists because it offers a small addition to the security model by allowing you the option not to allow the remote root user to have the same all-powerful access rights as the server's root user, by remapping it to a different UID. However, it is usually more convenient to allow it.

Due to the complex dependencies between the components, and the large number of them, it is usually more convenient to reboot the system than to start each of the services individually.

### Mounting NFS shares on the client

On another machine (or even on the local one), NFS shares can be mounted from the command line with a command such as the following:

```
# mount -t nfs server.example.com:/home /mnt
```

Depending on the client system (e.g. Linux or FreeBSD), it may mount the share using NFSv3 or NFSv4.

To mount a system with NFSv3 specifically in FreeBSD, the command would look like this:

```
# mount -t nfs -o nfsv3 server.example.com:/home /home
```

and correspondingly, to mount it using NFSv4:

```
# mount -t nfs -o nfsv4 server.example.com:/home /home
```

Note that when using NFSv4, the domain names of the client and the server need to match. In FreeBSD, this is configured with the `-domain` argument to the `nfsuserd` service. In Linux, it is governed by `/etc/idmapd.conf`. However, this facility is far from perfect.

To mount the NFS file system on boot, a line such as the following needs to be added to `/etc/fstab`:

```
server.example.com:/home /mnt nfs rw,nfsv4 0 0
```

In addition to the “rw” and “nfsv4” options, it is usually advisable to add three more options: “soft”, which makes IO operations on the client fail if the server is unavailable, “intr”, which makes IO operations interruptible (again, useful if the server is unresponsive), and “tcp”, which forces the NFS client to use TCP to mount the share (the alternative being UDP, which in modern systems may actually be slower than TCP, and is often much less convenient).

## Firewalling Your Server with ipfw

Network firewalls block (prevent) certain types of network packets from arriving at certain system services. Modern security practice is actually to block ALL packets except a very small set of well-chosen ones, in order to minimize the “attack surface”, or the number and the scope of services which are exposed to the network.

The packets can be blocked on several levels of the network stack. At the lowest practical level (L2), you could block packets with certain MAC addresses. It is much more usable to block packets coming or going to certain IP addresses (L3), and it is even more usable to block them based on the properties of TCP connections, such as ports or established TCP sessions (L4+). Finally, there are firewalls which perform deep packet inspection and operate based on the application data which is transported through TCP connections (L7).

The introduction of firewalls into a network system requires careful planning because it is very easy to unintentionally block certain services or systems, even from the administrators.

### Stateful firewalls

Old-style firewalls operate exclusively based on the static information available in individual packets, and because of that they are called “stateless”. For example, such firewalls might block packets to certain IP addresses, or to certain TCP ports on a packet-by-packet basis. However, slightly more advanced protocols, such as TCP, have implied session state attached to network packets, which is maintained simultaneously on both parts of the connection. Based on what the state of the connection is, certain types of packets may or may not be acceptable at an endpoint. For example (and very simplified), TCP has network packets which are used to establish a connection, packets which carry data, and packets which carry data acknowledgements. It is a violation of the protocol to send packets which supposedly relate to a connection before (or after) this connection is properly established, or to send

acknowledgement packets for non-existing data packets. Firewalls which are aware of this and keep track of TCP state are called “stateful.”

Firewalls implemented as a part of operating systems (like ipfw) operate below the operating system’s regular TCP/IP network stack and, as such, must keep the same type of information as the network stack does, though not all of it. This means a slight duplication of data between the firewall and the regular network stack, but is necessary to prevent potentially malicious packets to reach this stack.

### How ipfw works

Ipfw is one of the firewalls available in FreeBSD by default. The other ones are pf and ipf.

The core concept of ipfw is a numbered list of firewall rules which is traversed in order for each packet received. Each packet is tested against the rules and if a rule applies, certain action is taken (for example, the packet is dropped or unconditionally accepted). This list is of fixed size and can contain up to 65535 numbered entries. The numbers assigned to individual rules have no meaning outside being used for sorting the rules.

Network packets are matched with the rules based on certain data they contain, such as MAC addresses, IP addresses, TCP ports, or even the package data, or certain information which is associated with them in addition to the data they contain, such as from which network interface have they been received, or the state of the TCP connection they are a part of.

The list of things ipfw can do with packages is quite long, and includes NAT and traffic shaping.

### Rule syntax

The ipfw rule syntax is human-readable, and generally follows this form:

```
<action> <protocol> from <source> [ to <destination> [recv
    | xmit | via <nic>] ] [ options ]
```

The most common actions are `allow` and `deny`. Protocol is usually `ip`, `tcp` or `udp`. Source and destination are IP addresses or one of the special keywords such as “any” or “me”, and the `nic` is the name of a network interface. The `options` depend on a specific protocol and can include keywords such as “keep-state” to enable stateful matching for this rule, or “setup” to only apply to TCP connection setup (SYN) packets.

### Enabling ipfw

Ipfw is not enabled by default. It is available as a kernel module which will be automatically loaded at boot time if

enabled in `/etc/rc.conf`. By default, `ipfw` will load with only a single rule loaded, which drops all network traffic. Because of this, we need to configure some sane default rules in order to access the system remotely (e.g. via SSH).

This can be done by creating a file named `/etc/ipfw.rules` with the following lines:

```
add 500 allow ip from any to any via lo0
add 600 deny ip from table(0) to me
add 1000 allow icmp from any to any
add 2000 allow tcp from me to any setup keep-state
add 2001 allow udp from me to any keep-state
add 4022 allow tcp from any to me 22 setup keep-state
add 4080 allow tcp from any to me 80, 443 setup keep-state
add 5000 allow ip from 192.168.1.0/24 to 192.168.1.0/24
add 65400 deny log ip from any to any
```

After this, the firewall can be safely enabled in `/etc/rc.conf` with the following lines:

```
firewall_enable="YES"
firewall_type="/etc/ipfw.rules"
```

The rules in the example list given above are:

- 500 – First, allows all IP traffic through the loopback interface. This is an early rule because of efficiency (to skip matching the localhost packages through all the other rules).
- 600 – Denies all packages whose IP addresses are found in table #0 (tables will be described later).
- 1000 – Enables ICMP unconditionally.
- 2000 – Enables establishing connections (“setup”) from all IP addresses assigned to local network interfaces (“me”) to any foreign address (“any”), with stateful packet inspection (“keep-state”).
- 2001 – Allow UDP packets from all IP addresses assigned to local network interfaces to any foreign addresses, and allow a short time to receive a response on the same port (“keep-state”).
- 4022, 4080 – Allow establishing TCP connections (“setup”) from any foreign address (“any”) to all IP addresses assigned to local network interfaces (“me”), on ports 22, 80 and 443, with stateful packet inspection (“keep-state”).
- 5000 – Allow all IP packets regardless of protocol or any other details from all addresses on a local network to all addresses on the same network. This also matches all addresses on local network interfaces. This rule effectively opens the system entirely to the network traffic on the local subnet.

- 65400 – Denies all other network packets and logs them to syslog.

In addition to being read from the configuration file at boot time, rules can be loaded into the firewall by using the “`ipfw`” command as the root user, for example:

```
# ipfw add 4025 allow tcp from any to me 25 setup keep-state
```

This rule will open up the TCP port 25, used for SMTP, to any client on the Internet.

The rules are not automatically saved to the `/etc/ipfw.rules` file. This file needs to be maintained manually.

## Preventing SSH brute-force attacks

If your system has a routable IP address and is exposed to the Internet, you will probably find hundreds of opportunistic brute-force SSH attacks logged in the `/var/log/security` file daily. Such attacks usually try dozens of common usernames and passwords and move on when none of them succeed, but should nevertheless be blocked.

An easy way to block such attacks is by using the “`sshit`” program in the package of the same name. After installing it, edit the `/usr/local/etc/sshit.conf` file, and configure “`FIREWALL_TYPE=ipfw2`”. This configuration file also contains the parameters under which `Sshit` will block IP addresses.

`Sshit` works by parsing the system’s `auth.log` for certain types of records, and instructs the firewall to block appropriate IP addresses by putting the addresses into table #0. Tables in `ipfw` are lists of addresses which can be used in place of simple individual addresses, in which case the firewall action will apply to all addresses in the table.

---

## ABOUT THE AUTHOR

*Ivan Voras is a FreeBSD developer and a long-time user, starting with FreeBSD 4.3 and throughout all the versions since. In real life, he is a researcher, system administrator and developer, as opportunity presents itself, with a wide range of experience from hardware hacking to cloud computing. He is currently employed at the University of Zagreb Faculty of Electrical Engineering and Computing and lives in Zagreb, Croatia. You can follow him on his blog in English at <http://ivoras.net/blog> or in Croatian at <http://hrblog.ivoras.net/>, as well as Google+ at <https://plus.google.com/+IvanVoras>.*



## SharePoint is at the Crossroads — Which Way Will You Go?

**SharePoint in the cloud or on premises? Or both?** Come to SPTechCon Austin 2015 and learn about the differences between Office 365, cloud-hosted SharePoint, on-premises SharePoint, and hybrid solutions and build your company's SharePoint Roadmap!

For developers, the future means a new app model and new app paradigms. For IT pros and SharePoint admins, it's trying to retain control over an installation that's now in the cloud. For information workers and their managers, it's about learning how to work 'social.' But it's not for everyone.

Where do you need to be?

**The answer is simple:** SPTechCon Austin. With a collection of the top SharePoint MVPs and expert speakers, more than 80 classes and tutorials to choose from and panels focused on the changes in SharePoint, SPTechCon will teach you how to master the present and plan for the future.

**Migrate to SharePoint 2013! Prepare for Office 365!**  
**Build Your Hybrid Model!**



February 8-11, 2015  
Renaissance Austin Hotel

**80+ Classes**

**40+ Microsoft Expert Speakers**

**Get Your Texas-Sized Registration Discount—**  
**Register NOW!**

**[www.sptechcon.com](http://www.sptechcon.com)**

A **BZ Media** Event

SPTechCon™ is a trademark of BZ Media LLC. SharePoint® is a registered trademark of Microsoft.

# Python Flow Control Statements

Let's start with basic control flow statements. The first statement will be the if/elif/else. This is the most basic control flow statement you can have in Python.

---

## What you will learn...

- How Python initialises new objects.
- How to override Python's built-in methods and types.
- How to get an instance's attributes using the Python shell.

## What you should know...

- Python basics.
  - How to get data from a publicly accessible API.
- 

Here you will find the examples of the if statement. First you can see the full description of the statement.

```
if condition:
    statement
elif other_condition:
    statement
...
else:
    statement
```

Let's try to check if some number is even. The way to do it would be:

```
if number % 2 == 0:
    is_even = True
else:
    is_even = False
```

Another way of writing this code could be:

```
is_even = False
if number % 2 == 0:
    is_even = True
```

We could even do:

```
is_even = True if number % 2 == 0 else False
```

Or even simpler:

```
is_even = number % 2 == 0
```

All these lines are equivalent. They achieve exactly the same goal, which is to check if a number is even. Of course, this is a very simple example but you can use it for more complicated tasks, like:

```
processed_articles = slow_process(articles) if
    len(articles) < 10 else quick_process(articles)
```

## For Loops

Let's move into the more interesting control flow statements, which are *for*s and *whiles*. These are the statements that allow us to loop through iterables. An iterable is some object in Python that you can iterate through, like lists or dictionaries. The for loop syntax is:

```
for value in iterable:
    statement
```

Simple right? Let's try to do something more interesting with a list. We'll need the squared value for every natural number through 10.

```
squared = []
for n in range(10):
    squared.append(n**2)
```

The results for this operation will be `[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]` as expected. The range function is a great way of iterating through natural numbers. Another way of doing this loop could be:

```
squared = [n**2 for n i range(10)]
```

This is called a list comprehension and, although we are not covering them, we know that they are great and usually have better performance than for loops (when they can be used).

## Introduction to Python Programming Language

Python was created by Guido Van Rossum in the Netherlands. The language itself was created to be as simple as possible to read and use. Guido is still on the team of people who create Python and he is known in the community as the benevolent dictator for life (BDFL). The Python philosophy is summarised in the Zen of Python, which is a collection of guidelines that every Python core developer follows and every Python programmer should follow. The most important ones for people starting with Python are:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Readability counts.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Python is a language that is gaining a lot of popularity as a learning language because of its ease of use and its "batteries included" philosophy. These batteries mean that any Python standard distribution comes with a very powerful set of libraries to help overcome common, simple problems. For example, http connectors, sqlite (a relational database), csv readers, and a simple http server (actually called SimpleHTTPServer) are included, amongst many others.

The current version of Python is 3.4, but for this workshop we'll use version 2.7, which is in FreeBSD's repositories. Everything we do here also applies to Python 3.4.

Python is a very well documented language. You can check the official documentation that includes all the information you will need and also many tutorials you can try out.

## While Loops

The while loop is much like the for but you use a condition to decide whether to leave the loop or not. You read the while exactly as it's written: execute statement while condition is True.

```
while condition:
    statement
```

While loops are not used as often as for loops so just keep in mind that they exist.

## Lists (Slices), Dictionaries (Loop Over Items), Sets

Python has some very interesting and easy to use built-in data types. Lists are one of those and probably the one that you'll be using the most. To create a list, you have `articles = []`. This creates an empty list ready to be used. Lists do not require that every element be the same type. Any list can include numbers, strings, other lists, tuples, dictionaries... The most basic operations you can perform with a list are:

- append – add an element to the list.
- remove – remove an element from the list.
- index access – you can access any position on the list by index number.
- list slice – this is a very useful list operation that we'll go into in more detail.

Above I mentioned tuples. Tuples are basically a fixed size list. They can be used when you know that your list will always have the same size. This is good because tuples perform better than lists (but with the mentioned size restrictions).

## List Slices

Whenever you have a list, you might want to get a subset of it, or even access the last element. This is where list slicing comes into play. If you have a list with numbers from 0 to 10 but you only want the values from the 5th position to the last one, you can do:

```
>>> numbers = range(10)
>>> numbers
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> numbers[5:]
[5, 6, 7, 8, 9]
```

The same way you can get the element from the 5th position on, you can also get the first 5 elements:

```
>>> numbers[:5]
[0, 1, 2, 3, 4]
```

This is very useful when manipulating lists. Now let's imagine you need to know what the last element of the list is. You would usually do:

```
>>> numbers[len(numbers) - 1]
9
```

But Python allows you to do:

```
>>> numbers[-1]
9
```

This means that you can loop through your list in reverse order by making the indexes. This means that you can loop through the elements of your list in reverse. To do that you just start at -1 of the list all the way to the size of the list.

## Dictionaries

Dictionaries are another often used data type in Python. This is simply a hash table. Instead of having a number defining where your element is, you have a key. This key can be a lot of things, a number, a string, even a datetime object. For example:

```
>>> articles = {
... 123: "Article 1",
... 124: "Article 2"
...}
>>> articles[123]
Article 1
```

In this example, we are indexing each article to their unique identifier. If this was a list, the keys would have to be integers starting at zero (unless you wanted to fill all the indexes from 0 to 122 with None to be able to access the article with `articles[123]`).

There are some ways of looping through dictionaries. I'll focus on the most common.

```
>>> for key in articles:
...     print key, articles[key]
123 Article 1
124 Article 2
```

As you can see, iterating through the dictionary will iterate through its keys. You can then access the dictionary using the key. Another way to make this easier to read could be:

```
>>> for key, value in articles.items():
...     print key, value
123 Article 1
124 Article 2
```

In this case, we just use the `items()` method to create a list of key / value pairs so that the for loop can iterate through.

One thing you must keep in mind is that dictionaries are not ordered. This means that you cannot expect the loop to go through the dictionary in the same order that you added the elements. Python does have an `OrderedDict` which makes the dictionary stay in order but it performs worse than the normal dictionaries.

We will be using dictionaries later in this module and you'll have time to get familiar with them.

## Python Internals

Python is an Object-Oriented language and, as such, it deals with classes and class instances. If you are not familiar with Object-Oriented programming that's OK but there might be some definitions you might not know (like instance or instance variable). But I'll try to explain as we go.

## Classes and Object Instances

A class is an object that wraps a behavior. In the last model, we had the following code:

```
sample_csv_url = "https://raw.githubusercontent.com/pedroma/python-workshop/master/SacramentoRealEstateTransactions.csv"
response = urlopen(sample_csv_url)
f = open("SacramentoRealEstate.csv", "w")
f.write(response.read())
f.close()
```

The `open()` function returns a File object. The object is a File class instance and you can call methods on it (like the `write()` method in the line after).

Note that I called `write()` a method and not a function. This is because `write()` is a function that belongs to the File object and works within its context. If you just called `write()` without the

File instance, it wouldn't work because it needs to know where to write to.

Let's create a simple class and try this for ourselves.

```
class Point(object):
    x = 0
    y = 0
```

After defining this class, we can instantiate it (create an instance of this class by doing):

```
>>> p = Point()
>>> p.x
0
>>> p.y
0
```

Right now, we have a `Point` with coordinates (0,0). But there's nothing we can do to change it. Let's add some code that will allow us to create an instance with different values for `x` and `y`.

```
class Point(object):
    x = 0
    y = 0
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
>>> p = Point(x=3, y=5)
>>> p.x
3
>>> p.y
5
```

Good start. Let's try to understand what happened here. We defined a method inside `Point`. This is a special method. The `__init__` method is for class initialization and allows us to do some processing when a new instance is created. Our goal was to be able to set different values for `x` and `y` and it worked. The `self` parameter you see is so that we can access the current class and assign values to it. In other languages, you don't usually see this parameter (it is assumed to be there) but Python has the "explicit is better than implicit" philosophy.

### Everything is an Object (docs strings, getters, setters, override)

In Python, everything is an object. Functions, modules, even integers and built-in functions. Here is an example to show what I mean.

```
>>> i = 3
>>> i.bit_length()
2
```

If you want to know what a function does, you can ask the function for its `__doc__` attribute, which returns the documentation (provided the function has any). Let's ask Python what the `bit_length` function does.

```
>>> i = 3
>>> print i.bit_length.__doc__
int.bit_length() -> int
Number of bits necessary to represent self in binary.
>>> bin(37)
'0b100101'
>>> (37).bit_length()
6
```

As you can see, the `bit_length` function returns how many bits are needed to represent a number in binary. Most of Python's core functions are documented to help you when using them.

This "everything is an object" approach is very valuable and provides us with some very powerful features. Let's try doing something more advanced. Let's play with getters.

A getter is a special method that allows you to get an item from an object.

Let's improve Python's dictionaries. Initialize a dictionary and try to access a key that we haven't inserted yet.

```
>>> articles = {}
>>> articles["article1"]

_____
KeyError Traceback (most recent call last)
<ipython-input-10-59d0a8c1d287> in <module>()
--> 1 articles["article1"]
KeyError: 'article1'
```

When we try to do this, we get an exception. Let's create a dictionary that's doesn't throw one when we access a non-existent key. Because everything is an object, we can override Python's dictionary object and provide our own behavior. For example:

```
class MyDict(dict):
    def __getitem__(self, key):
        if key not in self:
            return None
        return super(MyDict, self).__getitem__(key)
>>> articles = MyDict()
>>> articles["article1"]
```

As you can see, now we don't get an error when accessing a key that doesn't exist. Of course, this is a very normal thing to want to do in Python and there is a shortcut for it.

```
>>> articles = {}
>>> articles.get("article1")
```

The `get` method from a dictionary is a "get or return None" (alternatively you can pass a second argument to replace the None (for example: "if you can't find the value return the string "No value")).

This is a very powerful feature because it allows you to do some very advanced stuff. For example, applying this to lists, you can build a spreadsheet where you do `sheet["A1"]` and internally you parse the "A1" to be the positions `[0][0]` on your list. You can check an example of that here.

Now let's play with setters. They follow the same rules as getters but for setting values. Let's imagine that for some reason when we add a number to a dictionary, we want to increment it by one. You could do it this way.

```
class MyDict(dict):
    def __setitem__(self, key, value):
        if type(value) == int:
            value += 1
        super(MyDict, self).__setitem__(key, value)
>>> articles = MyDict()
>>> articles["article"] = 1
```

```
>>> articles["article"]
2
```

As you can see, we tried to store a 1 but we stored a 2. This is an advanced feature and allows us to understand how Python works internally so we can take better advantage of the language.

### Practical example

*Let's do a slightly bigger example by using Twitter's API to get tweets. Twitter's API is a great source of information and easy enough to use that we can take advantage of it*

 Application Management



## Practical Python Workshop

Test OAuth

Details




Settings

Keys and Access Tokens

Permissions

### Application Settings

*Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.*

Consumer Key (API Key)	
Consumer Secret (API Secret)	
Access Level	Read-only ( <a href="#">modify app permissions</a> )
Owner	pedromcaraujo
Owner ID	

### Application Actions

Regenerate Consumer Key and Secret

Change App Permissions

### Your Access Token

*You haven't authorized this application for your own account yet.*

*By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.*

### Token Actions

Create my access token

Figure 1.

for this workshop. The main way to access Twitter is the Authentication system (OAuth2). We'll be using a Python package to make our work easier called Tweepy. Let's start off creating a virtual environment (or use the one created in the first module) and install the package Tweepy.

```
$ virtualenv python-workshop
(python-workshop) $ pip install tweepy
Downloading/unpacking tweepy
Downloading tweepy-3.1.0-py2.py3-none-any.whl
Downloading/unpacking requests-oauthlib==0.4.1 (from
tweepy)
Downloading requests_oauthlib-0.4.1-py2.py3-none-any.whl
Downloading/unpacking six==1.7.3 (from tweepy)
Downloading six-1.7.3-py2.py3-none-any.whl
Downloading/unpacking requests==2.4.3 (from tweepy)
Downloading requests-2.4.3-py2.py3-none-any.whl (459kB):
 459kB downloaded
Downloading/unpacking oauthlib>=0.6.2 (from requests-
oauthlib==0.4.1->tweepy)
Downloading oauthlib-0.7.2.tar.gz (106kB): 106kB
downloaded
Running setup.py (path:/home/pma/.virtualenvs/workshop/
build/oauthlib/setup.py) egg_info for package oauthlib
Installing collected packages: tweepy, requests-oauthlib,
six, requests, oauthlib
Running setup.py install for oauthlib
Successfully installed tweepy requests-oauthlib six
requests oauthlib
Cleaning up...
$
```

After this is installed, we need to get some API keys so we can use Twitter's API. To do that, just go to [https://](https://apps.twitter.com/app/new)

[apps.twitter.com/app/new](https://apps.twitter.com/app/new) (you'll need to be signed in to be able to register your app). Fill in the form (the application website is not really necessary, you can just put anything there).

After this is done, you should get sent to your application page: see Figure 1.

Your application is created and you have your consumer key and consumer secret to be able to tell Twitter this is the one making the request. But now we need two more keys to tell Twitter which user is making the request. Those keys are called "Access Tokens". To get them, just click on the "Create my access token" button at the bottom of the page. After you click it, you should get a new section on the bottom of the page like this one: Figure 2.

Awesome. At this point you should have all four keys needed to authenticate with Twitter's API:

- Consumer Key
- Consumer Secret
- Access Token
- Access Token Secret

At this point, we're ready to ask anything of Twitter. Let's start with some simple things.

Create a new Python file in your preferred editor and fill the following Python variables with it.

```
CONSUMER_KEY = ""
CONSUMER_SECRET = ""
ACCESS_TOKEN = ""
ACCESS_TOKEN_SECRET = ""
```

Now, with Tweepy installed, let's instantiate a Twitter API object and get it ready for requests.

## Your Access Token

*This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.*

Access Token	[REDACTED]
Access Token Secret	[REDACTED]
Access Level	Read-only
Owner	pedromcaraujo
Owner ID	[REDACTED]

Figure 2.

```
import tweepy
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
api = tweepy.API(auth)
tweets = api.home_timeline(count=5)
for tweet in tweets:
    print "User", tweet.user.name, "tweeted:", tweet.text
```

This small piece of code will get the five most recent tweets in our timeline and show the user who tweeted them as well as the tweet text.

If you check what a tweet is (what type of object it is), Python will tell you it's a "Status" object. That's because Tweepy's developers created a class to put all the status attributes in. If you want to know everything a Status object has, you have two main options. Either find Tweepy's source code and look at it or use the built-in `dir` function. You can check the `dir` function's documentation by running "`print dir.__doc__`" like you've seen earlier.

```
type(tweet)
tweepy.models.Status
dir(tweet)
['_class_',
....
'author',
'contributors',
'coordinates',
'created_at',
'destroy',
'entities',
'favorite',
'favorite_count',
'favorited',
'geo',
'id',
'id_str',
'in_reply_to_screen_name',
'in_reply_to_status_id',
'in_reply_to_status_id_str',
'in_reply_to_user_id',
'in_reply_to_user_id_str',
'lang',
'parse',
'parse_list',
'place',
'retweet',
'retweet_count',
'retweeted',
'retweets',
'source',
```

```
'source_url',
'text',
'truncated',
'user']
```

I've cut out some of the output from `dir` (you can ignore the attributes that start with an underscore, they are mostly for the object's internal use). As you can see, there's a lot of information you can get just from a single tweet.

## Conclusion

Now I leave you to explore Twitter's API. The same way you did a `dir` on a Status object, you can do it on anything to see their attributes, including the "api" object we instantiated earlier. Try it. Don't forget, you can use "`.__doc__`" to check what a method does (not all methods will have documentation but they're self explanatory).

Here are some fun calls you can try:

- `api.search_users(query)` – search users in the Twitter DB
- `api.get_status(tweet_id)` – gets a specific tweet by id
- `api.get_search(latitude, longitude)` – returns tweets in the vicinity of the coordinates you sent (try: `api.geo_search("37.391933", "-122.04375")` for tweets near Mountain View)

If you want to check Tweepy's full documentation just go to <http://tweepy.readthedocs.org/en/v3.1.0/index.html>.

---

## ABOUT THE AUTHOR

*Pedro Araujo graduated with a degree in Computer Science from Universidade do Minho. After this, he pursued his masters degree in Informatics choosing Parallel and Distributed Computing and Formal Methods. Pedro Araujo has diverse professional experience as he first worked at Euroutux, S.A. where he developed websites using Plone and Zope in addition to some system administration functions, like developing plugins for rails and Android development performed on a personal basis. Then, he moved to WeDo Technologies, a company that focuses in Revenue Assurance. There, he mainly worked on Java developments, Solaris and Linux servers. Presently, he is working at Network Locum, a promising startup and two additional projects, Novaweb and Attentive. All in all, he is a complete supporter of Open Source and his most preferred programming language is Python. The full online course on Python Programming is available on the BSDmag.org website.*

# Faster. Better. Reliable.

## Trusted by over 500 ISPs worldwide.

Hyper is the first multimedia cache fully developed in Brazil, by Taghos.

With Hyper, ISPs can save on network bandwidth while increasing content-delivery speeds, resulting in end-customer satisfaction.

### Features:

- 24x7x365 always-on support
- Active monitoring
- Automatic updates
- Appliance or license
- Easy deployment
- Configuration and reports via web interface



**Remote Install**  
Using your hardware

Model	Traffic	RAM	Cache	SSD
T15	Up to 15 Mbps	8 GB	1x 1 TB	-
T50	Up to 50 Mbps	8 GB	2x 1 TB	-
T100	Up to 100 Mbps	8 GB	2x 1 TB	1x 160 GB
T150	Up to 150Mbps	16 GB	3x 2 TB	1x 160 GB
T300	Up to 300 Mbps	16 GB	5x 2 TB	1x 240 GB
T500	Up to 500 Mbps	32 GB	7x 2 TB	1x 480 GB
T1000	Up to 1 Gbps	64 GB	10x 1 TB	1x 480 GB
T2000	Up to 2 Gbps	96 GB	24x 1 TB	3x 480 GB
T3000	Up to 3 Gbps	128 GB	32x 1 TB	5x 480 GB

Visit us at [www.taghos.com](http://www.taghos.com) and start saving bandwidth today!

# HardenedBSD, Always Ahead in Security

Previously, I focused on the Address Space Randomization Layout feature of the HardenedBSD project, handled by Oliver Pinter and Shawn Webb. HardenedBSD also has other features available and I'll try to describe all the features.

---

## What you will learn...

- About HardenedBSD and its features.
- The Address Space Randomization Layout feature.
- The arc4Random and ChaCha 20 features.

## What you should know...

- The OpenBSD functions.
  - Security and Kernel basics.
- 

Currently, FreeBSD uses the RC4 stream cipher for the arc4random family functions, both on the kernel and userland side. These functions serve many purposes; for example, on the kernel side, they allow the creation of proper randomized processes id, the stack protection canaries, and they are used by the HardenedBSD Address Space Randomization Layout as well.

On the userland side, openssh uses it widely and it is also used in the stack protection counterpart. It is generally an important piece of software.

Recently, in the last Hackfest (and previously in the last EuroBSDCon), Theo de Raadt discussed the arc4random OpenBSD's version and raised the need to move on from RC4 to a stronger stream cipher. Hence, the invention of ChaCha 20, implemented after the 5.5 release.

Subsequently, we decided to update HardenedBSD as well, on both the kernel and userland side. On the kernel side, the challenge was to keep it SMP safe while keeping the code change smooth and wise while on the userland side, the challenge was to update the fork detection. Indeed, when a fork is created, the reseeding is triggered. Usually, getpid function is used for this purpose but we thought there might be a better and more solid approach. M. Dempsey, an OpenBSD contributor, provided a new minherit flag, MAP\_INHERIT\_ZERO, to ensure that the memory map is properly zero'ed in this case. So, for HardenedBSD, a new INHERIT\_ZERO flag was added.

Related to this, a new system called *getentropy* was added, as well. Basically, it fills a buffer of randomized bytes with a maximum of 256 bytes. It serves more as an initial input for randomization rather than using it directly. Hence, for example, it can replace a couple of sysctl/KERN\_\*RND calls.

```
#include <unistd.h>
#include <err.h>

int
main(int argc, char *argv[])
{
    char buf[256];
    // errno can be set to EFAULT
    // or EIO (if more than 256 bytes are attempted)
    if (getentropy(buf, sizeof(buf)) != 0)
        errx(1, "getentropy failed");
    ...
    return (0);
}
```

## Some other libc functions

Again, we got inspired by OpenBSD and added some of their useful libc functions.

*gettablecount* which gives the number of file descriptors per process. It can be helpful alongside getdtablesize ...

```
#include <unistd.h>
#include <err.h>

#define FDRESERVE 5

int
main(int argc, char *argv[])
{
    ...
    if (getdtablesize() - getdtablecount() < FDRESERVE)
        errx(1, "running out of file descriptors");
    ...
}
```

*reallocarray* which checks some potential overflows (but does not zeroify)

```
#include <stdlib.h>
#include <err.h>

int
main(int argc, char *argv[])
{
    int *p, *q;
    // i.e same as realloc(NULL, 2 * sizeof(*p)); ...
    p = reallocarray(NULL, 2, sizeof(*p));
    if (p == NULL)
        errx(1, "reallocarray 1 failed");
    ...
    q = reallocarray(p, 10, sizeof(*q));
    if (q == NULL) {
        free(p);
        p = NULL;
        errx(1, "reallocarray 2 failed");
    }
    p = q;
    ...
}
```

A slightly different version of *strlcpy* is provided. *strlcpy* usually guarantees a zero at the end of the buffer, but the buffer does not sanitize the potential remaining bytes. So our version combines both *strlcpy* and *strncpy* advantages ... At the cost of a slight performance hit, only HardenedBSD, at the moment, does it.

```
#include <string.h>

int
main(int argc, char *argv[])
{
```

```
    char buf[10];
    // Will zeroify all the remaining bytes after the first
    // three.
    strlcpy(buf, "foo", sizeof(buf));
    ...
}
```

Finally, the *crypt API* was updated recently. Two new functions were added, *crypt\_newhash* and *crypt\_checkpass*. The latter provides an easy interface to test the validity of a password, while the first allows the creation of a hashed password. Once again, inspired by OpenBSD.

```
#include <crypt.h>
#include <err.h>

int
main(int argc, char *argv[])
{
    const char *passwd = argv[1];
    char hash[_PASSWORD_LEN];
    ...
    // errno can be set to EINVAL
    // Second parameter is the hash algorithm preference
    // the default is set if NULL
    if (crypt_newhash(passwd, NULL, hash, sizeof(hash)) != 0)
        errx(1, "crypt_newhash failed");
    ...
    // errno can be set to EACCES
    if (crypt_checkpass(passwd, hash, sizeof(hash)) != 0)
        errx(1, "crypt_checkpass failed");
    ...
}
```

## Conclusion

Summing up, you read about the most useful features available in the HardenedBSD project, as well as having the chance to see what new functions were added in the latest versions. I hope that you like this project and it will seem very familiar to those who like OpenBSD and work with it.

---

## ABOUT THE AUTHOR

*David Carlier has been working as a software developer since 2001. He used FreeBSD for more than 10 years and, starting this year, he became involved with the HardenedBSD project and performed serious developments on FreeBSD. He worked for a mobile product company that provides C++ APIs for two years in Ireland. From this experience, he became completely inspired to develop on FreeBSD.*

# Getting to Grips with the Gimp – Part 10

In the final part in our series on the Gimp we will wrap up and take a look at how to further improve your Gimp experience.

## What you will learn...

- How to manipulate images like a design pro

## What you should know...

- General PC administration skills

Over the past 9 articles, we have covered pretty much all the basic and intermediate skills required to use the Gimp effectively. We will now look at some of the softer skills and additional resources to improve your graphic design capabilities.

## Work-flow

Under pressure of a deadline? Not sure of the result you will achieve by applying a particular filter? While Ctrl Z will get you out of many sticky situations, saving regularly in XCF will prevent you from serious frustration. While hardware and software had improved greatly over the years, it is easy to get “drawn in” to the creative zone and when that keyboard or PC locks up a backup is essential. Remember also, that an exported

image e.g. a JPG will not hold Gimp specific data such as layers etc.

If you are processing a lot of images e.g. for a website, keep a copy of the image masters in a separate directory in case you inadvertently overwrite one of the images. This is useful where you manipulate an image and find something missing or wrong in the final product. It is easy



to miss something when you are looking at it for a long time, and often these little errors will not show up until the last moment.

### Tuning and customising the Gimp

Depending on the type of workload you anticipate you will be performing, you may be required to perform a number of repetitive tasks. For example, in web design, often images need to be scaled to a certain size. Gimp provides extensive hooks for key bindings that can be modified via edit → keyboard shortcuts. For instance, binding Ctrl Alt R to the scale menu item will allow the quick resizing of images to a desired size. Once the size has been set, this will be repeated each time Ctrl Alt R is pressed.

Loading, importing and designing your own brushes, paths, gradients and patterns is simplicity itself, just right click in the white space in the toolbox area. You can then reuse these as desired.

There are countless additional resources on the web, deviantart.com being amongst one of the best.

### Working with a design brief

Apart from getting inspiration, the most difficult part of design is getting the idea and concept out of the clients head into a format you can translate into an image or graphic. Don't be surprised if you need to have 2 or more passes until they are happy. I always start with 3 mock ups using different styles and moods to try and gauge what is re-

quired. If the client really wants orange and blue text on a green speckled polka-dot background so be it, just be thankful it is not your company logo!

Ultimately, beauty is in the eye of the beholder, and sometimes going against the grain does work. Personally, I find when I hit that “wow” moment when it just seems right (often without being able to qualify exactly why). The whole concept has to fit, culturally as well as how the message is communicated. People read different things into different images – one landing page I did for a website was approved by the communications manager and we both agreed the imagery was powerful, stunning and got the point across extremely well, while others were offended. Ultimately, you will never please 100% of the people 100% of the time. The occasional bit of controversy though is good if it raises the profile of the subject.

### Writing your own Gimp plug-ins

If you are a competent C programmer, the Gimp can be easily extended with a few lines of C. For more information, see <http://developer.gimp.org/writing-a-plugin/3/index.html>.

### Automated versus manual

There is a plethora of plug-ins and filters etc. available for the Gimp, many of them mimicking the capabilities of Photoshop. However, part of the fun (and the learning experience) is developing the skill of knowing what looks good and how to make a good image outstanding. While an automated filter or plug-in may scratch that itch for instant gratification, understanding the underlying mechanics of how the image is transformed can be of great benefit.

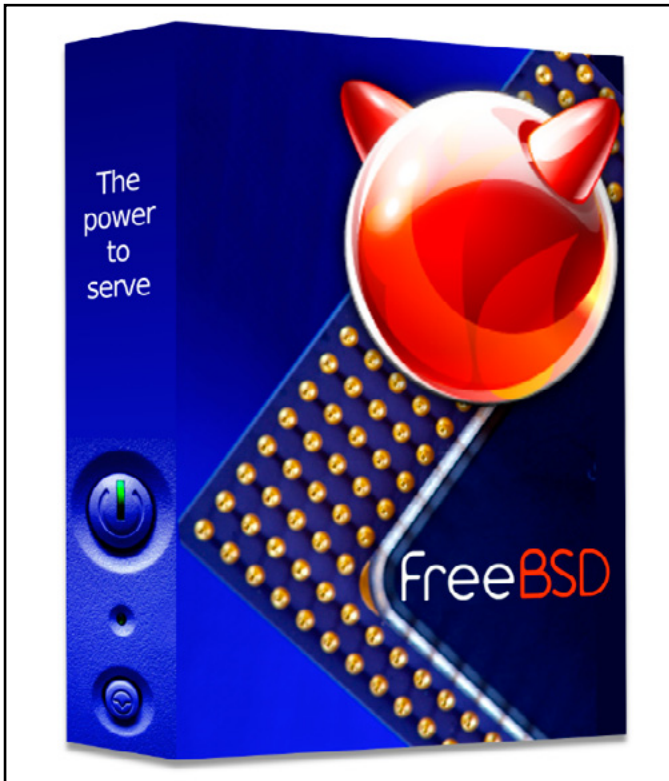




Many of the most stunning effects are achieved by applying multiple processes and manipulating many layers and selections. Don't be afraid to experiment, keep a notepad handy of the processes you have used and how you have arrived there.

## Your work environment

If you are serious about graphic design you will need a decent monitor, graphics card, a way to perform colour management and possibly a graphics tablet. Colours



on a LCD display differ wildly from that on an LCD display, and colours for print appear different from those on screen. Lighting is also important, glare and fluorescent light can make it difficult if not impossible to work accurately over long periods of time.

## Artwork and resources

While there is nothing new under the sun, just grabbing images off Google for graphic design purposes is considered very bad form in professional circles. Either use a professional stock agency such as Istock or material with a Creative Commons licence. Better still, take your own photographs. This is particularly important when performing graphic design for third parties, as their reputation could be at stake.

Inspiration is another matter entirely. Few artists would be narrow minded enough to complain if a particular technique is copied, and will probably look upon your endeavours as a compliment particularly if attributed. Getting inspiration is often the hardest part of design, so always be on the look out for new ideas and try to envisage how the designer has built the resulting image.

## Design tutorials

If you cannot be a good example you will have to be a terrible warning. Have a look at <http://thatcaption.com/25-photoshop-fails> or Google Photoshop fails to learn from the mistakes of others. Again, it is always easy to miss something, so an extra pair of eyes is always helpful to spot glaring errors in the final proof. Be conscious though that beauty is in the eye of the beholder. For an in-depth set of tutorials on graphic design, have a look at <http://www.lynda.com/Design-training-tutorials/40-0.html>. This covers techniques such as composition, typography, colour and logo design.

## And finally

I hope you have enjoyed this series on the Gimp. It is one of my most treasured Open Source programs due to its reliability, flexibility and the fact that it has never let me down. Experimentation is the key, and I hope you enjoy working with the Gimp as much as I have.

---

## ROB SOMERVILLE

*Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.*

# Great Specials

On FreeBSD® & PC-BSD® Merchandise

Give us a call & ask about our  
**SOFTWARE BUNDLES**

**1.925.240.6652**

**\$39.95**

FreeBSD 9.1 Jewel Case CD Set  
**or** FreeBSD 9.1 DVD

**\$29.95**

PC-BSD 9.1 DVD

**\$49.95**

The PC-BSD 9.0 Users Handbook  
PC-BSD 9.1 DVD



**\$99.95**

The FreeBSD CD **or** DVD Bundle

Inside each CD/DVD Bundle, you'll find:  
FreeBSD Handbook, 3rd Edition  
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide  
FreeBSD 9.1 CD or DVD set  
FreeBSD Toolkit DVD

*Stylish Dress Attire*  
Look Your Professional Best



*Comfy Apparel*  
Stay Warm in Zip Ups & Pullovers

*T-Shirts*  
Lots of Styles to Choose From

## FreeBSD 9.1 Jewel Case CD/DVD.....\$39.95

CD Set Contains:

- Disc 1** Installation Boot LiveCD (i386)
- Disc 2** Essential Packages Xorg (i386)
- Disc 3** Essential Packages, GNOME2 (i386)
- Disc 4** Essential Packages (i386)

FreeBSD 9.0 CD.....\$39.95

FreeBSD 9.0 DVD.....\$39.95

## FreeBSD Subscriptions

Save time and \$\$\$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.1.....\$29.95

FreeBSD Subscription, start with DVD 9.1.....\$29.95

FreeBSD Subscription, start with CD 9.0.....\$29.95

FreeBSD Subscription, start with DVD 9.0.....\$29.95

## PC-BSD 9.1 DVD (Isotope Edition)

PC-BSD 9.1 DVD.....\$29.95

PC-BSD Subscription.....\$19.95

## The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide).....\$39.95

The FreeBSD Handbook, Volume 2 (Admin Guide).....\$39.95

## The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes).....\$59.95

The FreeBSD Handbook, Both Volumes & FreeBSD 9.1.....\$79.95

PC-BSD 9.0 Users Handbook.....\$24.95

BSD Magazine.....\$11.99

The FreeBSD Toolkit DVD.....\$39.95

FreeBSD Mousepad.....\$10.00

FreeBSD & PCBSD Caps.....\$20.00

BSD Daemon Horns.....\$2.00



*Bundle Specials!*  
Save \$\$\$

*Just Plain Fun*  
Mousepads & Novelty Horns



BSD Magazine  
Available Monthly



For even **MORE** items  
visit our website today!

**www.FreeBSDMall.com**

**“If you’re moving information into the cloud, it just seems to me that all kinds of nasty activity could go on in there. I would take a Missouri approach and say – prove it to me, show it to me – how it’s more secure”.**

**A**ir Force General John P. Casciano (a former director of intelligence, surveillance and reconnaissance air and space operations for the US-AF) said “If you’re moving information into the cloud, it just seems to me that all kinds of nasty activity could go on in there. I would take a Missouri approach and say – prove it to me, show it to me – how it’s more secure”. With increasing pressure on budgets and resources, more and more organisations are looking towards moving their IT operations to the cloud. Is this a genuine dawn of a new technical revolution or are we potentially facing a major crisis further down the road?

Talking to an external consultant this week I raised my concerns about the thorny question of how – from an operational perspective – there is often a major disconnect between IT and senior management. Moving on from there, the subject of the cloud came up and I was astounded by the response when I expressed my doubts about the viability of the cloud, especially where security and confidentiality was paramount. To précis, the response was basically “If the government says it is OK and secure they can carry the can if the wheel falls off”. While I admire the level of pragmatism in bolstering the latest current management thinking, this confirmed to me once again that a) Technological hype will always trump common sense and b) in the relentless pursuit of efficiency and cost savings he who ignores the adage “Penny wise pound foolish” will eventually suffer both capital and reputational loss.

Both Microsoft and IBM performed a minor miracle in the 1980’s in democratising information technology – the end user was in control (albeit to a degree and at a cost) that was impossible under centralised, mainframe big-iron. Ironically, at board level exactly the same arguments were used then as are now regarding the cloud – you don’t want your organisation held hostage by a bunch of mission critical “specialists” that might

want better pay or conditions, or heaven forbid more investment in technology or infrastructure. “Thin the herd” was the cry, and as a result the IT industry fractured and spawned a plethora of roles but all that happened in reality was a transference of control to outside the organisation and a corresponding decrease in efficiency and customer service values. It is a lot easier to walk into IT and ask a favour than logging an external help-desk call and submitting yourself to the humiliation of a rigorously enforced Service Level Agreement. Call me old school, a dinosaur – I care not. I worked in IT before the SLA and the dreaded words “Expectations management” were *de rigour*. Customer service was IT policy and we were only happy when our customers were happy. Everyone was working for the same organisation with the same goals, priorities and corporate identity. Now we have the scenario where developers, system admins, project managers *et al* are external resources and in the typical scenario the vision is that market forces will prevail by bringing more efficiency, cost effectiveness and economies of scale to the table. Alas, all this fragmentation has wrought has been increased costs, deteriorating communications, lack of creativity and ingenuity and a “one-size fits all” mentality that has turned IT from a colourful exciting career providing solutions and service to a bland bureaucratic fire-fighting exercise or worse still, being in the role of consultant where by the very insecurity of the job itself means that you are there to provide what the client wants rather than dare to rationally debate what is the best solution. I have lost count of the number of freelancers who have said to me off the record “I know it is a bad solution, I wouldn’t do it myself, but it is what they hired me to do”.

As far as any modern organisation is concerned, IT holds a very intimate and critical role in respect of how it performs. However, this is no excuse to place IT on

a pedestal. If I was CEO of a company, my first concern would not just be of one efficiency, but of adding value and growth as well. Critically though, I would understand that success is based not just on tangibles like the balance sheet, but the many subtle currents that are invisible like synergy, personal chemistry, teamwork, relationship and vision etc. These are the invisible drivers of success, and are part of the hard to quantify metric that turns an organisation from good to great. Ultimately though, it comes back to power. Success is often at the hand of the benevolent dictator. As organisations have grown larger, like the IT industry itself they have fragmented more and more with the creation of specialised roles such as HR, Accounts, Health and Safety etc. While it is undeniable that medium and large sized organisations need these departments, the unforeseen consequence of this is not just the delegation of power, but further disconnect and division within the organisation itself. So rather than promoting efficiency, the CEO is ironically held hostage to departmental silos and the organisation becomes politicised, institutionalised and inflexible. Inter-departmental rivalry becomes a matter of corporate survival, and rather than focusing on the customer, the problem becomes the lack of cohesive leadership and vision as everybody is working in isolation. The IT parody about "Herding cats" has become the corporate meme.

The cloud is meant to be a part of the solution to this conundrum, as everyone will have a single view of the organisation, their customers etc., available from every device 24/7. However, unless the culture of the organisation is mature, well developed, accepted, agreed to and understood, there will always be a window of opportunity for the unethical, exploitative and opportunist to leverage and distort an organisations' values to their own end. To quote Casciano, show me where having additional layers of management, infrastructure, legislation, personnel, policy and culture increase security. I must admit here to using creative licence. While Casciano was probably referencing security in terms of black hat hackers, spies and trouble-makers, I prefer to use the term "secure" in a much more holistic sense.

While the cloud is great for flexible processing resources and accessing non-critical data, any organisation considering implementing an IT strategy where core business is based in a public cloud without con-

siderable redundancy and professional legal advice really needs to think more rigorously. A private cloud is a much better risk, but then the cost potentially rises way above the utilitarian public offering – cheap "everything" due to economies of scale. Those old enough to remember the first generation of ISP's will remember the tension between cost of provision and virtualised web servers, and the resulting flight of mission critical applications away from virtual hosting to dedicated servers once the developers or architects realised there was an issue with scalability and performance. With the cloud, we have kicked the problem a bit further down the road, and it will be the SaaS or IaaS provider who will have to deal with the issue and inevitably will hold the better commercial hand. Already we are seeing dissatisfaction with spiralling costs and excessive downtimes due to centralised failure.

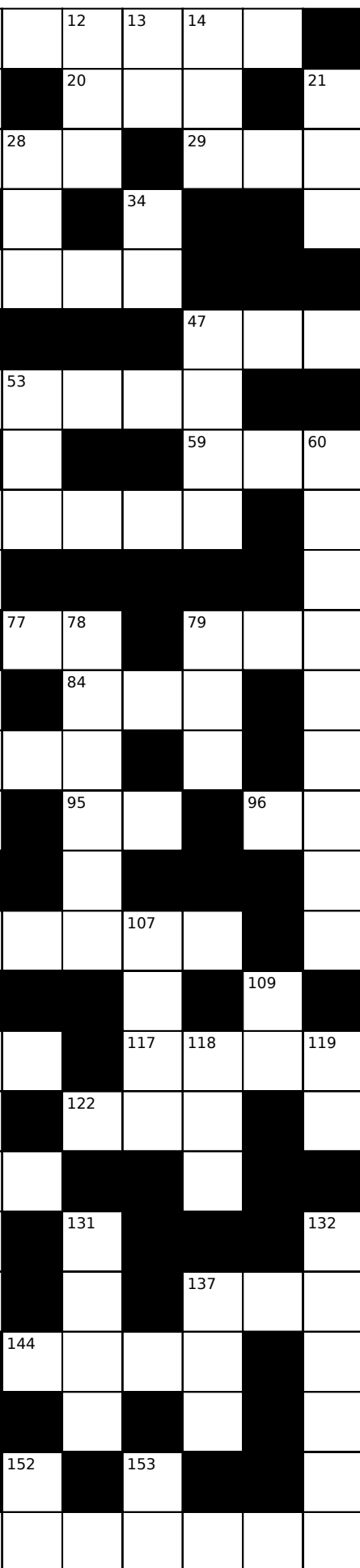
Any professional gambler will tell you of the need to spread risk. Placing all of one's eggs in a very public basket controlled by a global brand name whose sole unique selling point is trust – in my opinion constitutes a poor commercial decision. As we all know, when IT partners fall out often the only redress is often through the courts – and this is my biggest concern about the cloud. In a globalised society, who knows the legal incorporation of a local office. This might be fine for US corporations, but in Europe and elsewhere it is a minefield. Recently, a US magistrate judge ruled that Microsoft had to comply with a warrant asking for data held on their servers in Ireland. Microsoft is currently fighting this, and potentially this could end up with a major spat between US and EU courts over jurisdiction. Add to this all the additional points of failure, the ultimate loss of control, and I can see a few deeply embarrassed CEO's lining up to take the walk of shame. Have a great 2015 and for those that don't get where I am coming from, I shall finish with one word. Sony.

---

### ROB SOMERVILLE

*Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.*

	1	2		3	4		5			6		7		8	9	10		11
				15			16				17	18			19			
22							23	24						25	26			27
		30			31					32				33				
35							36	37					38				39	
		40		41		42	43				44		45				46	
				48						49					50	51		52
		54				55			56					57		58		
61					62			63					64					65
		66	67	68			69				70	71						
72								73			74				75		76	
							80			81							82	83
85						86				87					88			89
				90	91		92						93	94				
		97								98			99				100	
101				102				103						104		105		106
												108						
110				111			112			113		114			115			116
			120														121	
		123											124					125
				126						127		128				129	130	
	133	134					135								136			
138					139				140				141	142				143
145				146								147				148		
			149						150	151								
154										155					156			



## Across

- 1 SSL 3 man in the middle attack
- 5 Change and print terminal line settings
- 8 Block user or IP address from network
- 11 Remove, often forever
- 15 Abbreviation – Spanish dictionary
- 17 A very reliable file-system
- 19 A Christmas alcoholic drink used to install BSD applications
- 20 Fast paged mode memory, often used in servers
- 22 Quit Perl script and display a message
- 23 Decimal 255 in Hex
- 25 Conditional statement in many languages
- 27 Male goat used in computers
- 29 Boot sector at the very beginning of partitioned disk
- 30 Widely-used utility to keep copies of a file on two computer systems the same
- 32 If you don't read the documentation, the third letter may be an obscene word
- 33 Samba application layer protocol
- 35 Chipset manufacturer that should should make routers instead?
- 36 In assembler, store contents into an addressed location
- 38 As 23 Across
- 39 Ubuntu, Slackware is a type of this but \*BSD is not
- 40 Your CPU will contain quite a few of these
- 45 Common Unix editor
- 46 Official name for the One Laptop Per Child device
- 47 Application-layer protocol for Internet Telephony
- 48 Gas found in original computer display digits
- 49 Educational talk
- 50 Supply current that is not direct
- 52 In programming, a self-contained entity that consists of both data and procedures
- 54 The grandmother of non-numbers?
- 55 Intelligence that is not genuine
- 58 The programming environment for 52 across
- 59 Well known UPS manufacturer
- 61 Part of the Snort logo
- 62 A small green vegetable embedded in your pupil or a way for lawyers to make money perhaps?
- 63 You have this, your server doesn't.
- 64 Last term in the UK phonetic alphabet
- 65 A snake like digital circuit that performs addition of numbers
- 66 In Microsoft servers, a unique alphanumeric character string that identifies each operating system and each user
- 70 In programming, the first step before you read the contents of a file
- 72 Generate an image by tracing the path of light through pixels in an image
- 74 As 54 across
- 75 Older disk drive interface format
- 77 A piece of transparent plastic and foil often used as a mug coaster
- 79 German enterprise business applications based on plant contents?
- 80 13th letter of the Greek alphabet
- 81 1950's computer language
- 82 +13 – Usenet equivalent of a magazine printing the answer to a quiz upside down
- 84 Microsoft object embedding
- 85 Dull effect on most keyboards and computer cases
- 87 Polite terminal programs will say this when quitting
- 89 Type of processor based on upper limbs?
- 90 Bash, C and Korn
- 93 Outdated PC bus architecture
- 95 Decimal 173 in Hex
- 96 Decimal 238 in Hex
- 97 A child's toy hanging from the ceiling, most devices are this these days
- 98 A collaborative enterprise often requiring a manager
- 101 Median value of 50 across
- 102 The time co-ordinate everywhere
- 103 Form method used in HTTP
- 104 In systems design, allowing for the unexpected
- 108 MS outlook or Unix Mutt
- 110 Service oriented design pattern

111 You will need this to take your server apart  
 113 A common programming method to collect errors  
 115 Another version of 103 across  
 116 Original IBM PC architecture  
 117 The original type of Unix or mainframe terminal  
 120 A unit of current equal to one thousandth of an ampere  
 121 SMTP 250 reply code  
 122 Name for a security system related to clue for 61 across  
 123 Microsoft binary  
 125 Information technology  
 126 Part of IBM's now elderly token network  
 127 Mass loading of tapes or CD's or what you are reading now  
 133 A remote HTTP server  
 135 Abbreviation for an application front end  
 136 Generic name for the primary field in a database table  
 137 Abbreviation for Teletype  
 141 Many of 133 across would be found here  
 144 Slang term for motherboard  
 145 Abbreviation for random  
 146 No dungeon is complete without this  
 147 Function call to get the length of a string  
 148 Method for addressing memory  
 150 A thousand or so megabytes  
 154 Base-2 numeral system  
 155 8 of these in a byte  
 156 Encoding information using fewer bits than the original representation

#### Down

2 OpenBSD, NetBSD and DOS are these  
 3 Prevent access to a system  
 4 Unix list command  
 6 Last 2 letters of the alphabet  
 7 Abbreviation of 2 down  
 9 Abbreviation for a wireless access point  
 10 Derogatory term for a newcomer  
 11 RS232 control signal on pin 20  
 12 Abbreviation – Lindermann electrometer  
 13 Unix editor  
 14 The cat in Apache's servlet container  
 16 As 9 down  
 18 Oldest request is handled next in the queue or stack  
 21 Rights management that is not analogue  
 22 A programmer or coder  
 24 Computer software that can be classified as both free software and open-source software  
 26 Non-profit organization founded by Richard Stallman  
 28 Notation that describes rules and structures for representing encoding transmitting and decoding data  
 31 Number of pins in a VGA connector  
 32 Superuser account  
 34 DNS email record  
 38 Closing statement for 'if' in shell  
 39 In programming, go back and do it again  
 41 Abbreviation for Artificial Neural Network  
 42 Protocol specification for exchanging structured information  
 43 Common file containing startup parameters  
 44 Do not write if flagged with this  
 47 Another term for asterisk  
 51 To add  
 53 Another editor which uses the s-lang library  
 56 IP test if another device is there  
 57 SCSI Unit identifier  
 60 Individual electronic part of 144 across  
 64 Accelerated Turing machine

65 DNS Authoritative answer  
 67 As 125 across  
 68 Drupal inventor  
 69 Unix binaries directory  
 71 Memory section or printer output  
 73 Without value, consequence, or significance  
 76 Hopefully what you design in Gimp  
 78 A unique name that identifies an internet resource  
 79 Another editor – possibly found close to a river?  
 81 ----, retry, ignore  
 83 To break into a network cable  
 85 Bill Gates' empire  
 86 Hook and eye material used to secure cables and space-suits  
 88 Unix command akin to a middle eastern fruit  
 91 You want this on your website but not physically  
 93 85 across browser  
 94 Collect data from screen  
 98 Class of Intel CPU  
 99 What happens when JMP is executed by an assembler  
 100 Unix command to change user  
 105 You don't want this inside your server  
 106 DNS nxdomain abbreviation  
 107 Father of relationship database programming  
 109 Abbreviation for quick chat program  
 112 132 down is this  
 114 See 9 down  
 116 Utility to destroy an X-window  
 118 Replaced serial ports in most PC's  
 119 Decimal 191 in Hex  
 120 See 34 down  
 124 Gaseous product from laser printers  
 128 Graphics file format standard  
 129 Network card  
 130 Another editor  
 131 Server footwear on startup  
 132 Snake like programming language  
 134 CRT or LCD menu  
 137 Unix utility to show running processes  
 138 Linux bootloader found in rotten fruit  
 139 Supercomputer manufacturer  
 140 Graphics file format standard  
 142 Binary 1 in octal  
 143 Common term for Apache and MySQL running on a Redhad box  
 149 A hardware interface between a computer and an Integrated Services Digital Network line  
 151 Decimal 187 in Hex  
 152 Decimal 254 in Hex  
 153 Abbreviation for a system composed of people and computers that processes or interprets information

# Big Data Gets Real in Boston!

**People are talking about  
BigData TechCon!**



**"Big Data TechCon is a great learning  
experience and very intensive."**

—Huaxia Rui, Assistant Professor,  
University of Rochester



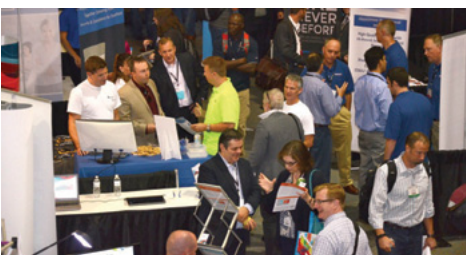
**"Get some sleep beforehand,  
and divide and conquer the packed  
schedule with colleagues."**

—Paul Reed, Technology Strategy & Innovation, FIS



**"Worthwhile, technical, and a breath of  
fresh air."**

—Julian Gottesman, CIO, DRA Imaging



**"Big Data TechCon is definitely worth the  
investment."**

—Sunil Epari, Solutions Architect, Epari Inc.

## BigData TECHCON

**April 26-28, 2015**

Seaport World Trade Center Hotel



**Choose from 55+ classes and tutorials!**

**Big Data TechCon is the HOW-TO technical conference  
for professionals implementing Big Data solutions  
at their company**

**Come to Big Data TechCon to learn the best ways to:**

- Process and analyze the real-time data pouring into your organization
- Learn how to extract better data analytics and predictive analysis to produce the kind of actionable information and reports your organization needs.
- Come up to speed on the latest Big Data technologies like Yarn, Hadoop, Apache Spark and Cascading
- Understand HOW to leverage Big Data to help your organization today








**[www.BigDataTechCon.com](http://www.BigDataTechCon.com)**



**NET OPEN SERVICES** IS AN APPLICATION HOSTING COMPANY FOCUSED ON OPEN SOURCE APPLICATIONS MANAGEMENT IN HIGH AVAILABILITY ENVIRONMENT.

**NET OPEN SERVICES** IS PROUD TO PROVIDE A HIGH QUALITY SERVICE TO OUR CUSTOMERS SINCE 10 YEARS.

OUR EXPERTISE INCLUDES:

-  CLOUD COMPUTING, PUBLIC, PRIVATE AND HYBRID CLOUD MANAGEMENT (OPENSTACK, CLOUDSTACK, RED HAT ENTERPRISE VIRTUALIZATION)
-  REMOTE MONITORING AND MANAGEMENT 24/7
-  NETWORKING AND SECURITY (OPEN BSD, IP TABLE, CHECKPOINT, CISCO,...)
-  OS AND APPLICATION MANAGEMENT (FREE BSD, OPEN BSD, SOLARIS, UNIX, LINUX, AIX, MS WINDOWS)
-  DATABASE MANAGEMENT (ORACLE, MYSQL, CASSANDRA, NOSQL, MS SQL, SYBASE...)
-  MANAGED HOSTING IN CARRIER CLASS DATA CENTERS
-  DISASTER RECOVERY



WE PROVIDE SERVICES IN EVERY STEP OF THE PROJECT LIFE, DESIGN, DEPLOYMENT, MANAGEMENT AND EVOLUTIONS. **NETOPENSERVICES** TEAM INCLUDES EXPERIENCED LEADERS AND ENGINEERS IN THE INTERNET SERVER INDUSTRY.

OUR TEAM HAS 15 YEARS OF EXPERIENCE IN DEVELOPING INTERNET INFRASTRUCTURE-GRADE SOLUTIONS AND PROVISIONING INTERNET DATACENTERS AND GLOBAL SERVICE NETWORKS TOGETHER.

WE OFFER EXCEPTIONAL HARDWARE SUPPORT AS SOFTWARE SUPPORT ON UNIX/LINUX AND OPEN SOURCE APPLICATION. **NETOPENSERVICES** DELIVERS THESE CUSTOM-BUILT LINUX AND UNIX SERVERS, AS WELL AS PRECONFIGURED SERVERS AND SCALABLE STORAGE SOLUTIONS, TO OUR CUSTOMERS. WE ALSO OFFER CUSTOM DEVELOPMENT AND ADVANCED-LEVEL UNIX/LINUX CONSULTING SOLUTIONS.

# Meet the Developer-Friendly Payment Solution



## 3 easy steps to optimized checkouts:

1

### Create the checkout page

With Gate2Shop, you can optimize your payment pages by using ready-made templates or by customizing payment pages to your site look and feel.

2

### Test and optimize

An effective payment page variant testing tool, A/B Testing helps you gain insight into user behaviour, increase payment conversion in the short and long term.

3

### Accept payments worldwide

With dozens of alternative and local payment methods offered in multiple currencies, the personalized checkout allows you to reach users from all around the world.

✓ Easy integration   ✓ Cross-platform   ✓ Secure



Call for a free consultation: +44 20 3051 0330

[www.g2s.com](http://www.g2s.com)